

Ederson Torresini

**COLMÉIA: uma Tecnologia Adaptativa
de Entrada de Dados ao Computador**

Florianópolis - SC

2004

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO**

Ederson Torresini

**COLMÉIA: uma Tecnologia Adaptativa
de Entrada de Dados ao Computador**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Edla Maria Faust Ramos

Florianópolis, Setembro de 2004

COLMÉIA: uma Tecnologia Adaptativa de Entrada de Dados ao Computador

Ederson Torresini

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Raul Sidnei Wazlawick

Banca Examinadora

Profa. Dra. Edla Maria Faust Ramos

Profa. Dra. Araci Hack Catapan

Prof. Dr. Leandro José Komosinski

Profa. Dra. Lúcia Helena Martins Pacheco

*Se as coisas são inatingíveis... ora!
Não é motivo para não querê-las
Que tristes os caminhos, se não fora
A presença distante das estrelas!*
(Mário Quintana)

A todo aquele que crê na educação como prática da liberdade.

Agradeço primeiramente aos meus eternos professores,
desta e de outras vidas, Rodnei e Venina Torresini.

Edla Maria Faust Ramos, por ser luz e voz.

João Bosco da Mota Alves, mestre e amigo.

E a Carina Eligia Granemann, meu porto seguro.

Conteúdo

Lista de Figuras

Lista de Tabelas

Resumo

Abstract

1	Introdução	p. 15
1.1	Objetivos	p. 19
1.2	Justificativa	p. 20
1.3	Hipóteses	p. 23
1.4	Resultados Esperados	p. 23
1.5	Metodologia	p. 24
2	Os Meios de Entrada de Dados	p. 26
2.1	Os Meios Convencionais de Entrada de Dados	p. 26
2.1.1	O Teclado	p. 27
2.1.2	O <i>Mouse</i>	p. 28
2.2	Os Meios Alternativos de Entrada de Dados	p. 29
2.2.1	Ativação por Toque	p. 30
2.2.2	Ativação por Obstrução	p. 32
2.2.3	Ativação por Movimentos	p. 33

2.2.4	Ativação Indireta	p. 33
2.3	Classificação: Análise e Síntese	p. 35
3	Os Requisitos para a Escolha das Tecnologias	p. 36
3.1	Os Requisitos Básicos	p. 36
3.2	As Funcionalidades Adicionais	p. 37
3.3	A Tecnologia Base	p. 38
3.4	As Tecnologias Complementares	p. 40
3.4.1	Tela de Retorno Tátil	p. 40
3.4.2	Correção de Erros de Acesso	p. 40
4	O Periférico	p. 41
4.1	Os Componentes	p. 41
4.1.1	Dispositivo de Aquisição de Dados	p. 41
4.1.2	Comunicação com o Computador	p. 44
4.2	Correção de Erros de Acesso	p. 46
5	O Device Driver	p. 49
5.1	Recepção dos Dados	p. 49
5.2	Interpretação dos Dados	p. 51
5.2.1	Filtragem por Módulos Adicionais	p. 54
5.3	Consulta à Tabela de Eventos	p. 60
5.4	<i>COLMÉIA</i> : Independência da Aplicação	p. 63
6	A Aplicação Base	p. 65
6.1	Interface da Aplicação	p. 66
6.1.1	Janela de Criação	p. 67
6.1.2	Janela de Depósito de Perfis	p. 74

6.2	Considerações Finais	p. 76
7	Discussões e Conclusões	p. 78
7.1	Resultados Obtidos	p. 80
7.2	Trabalhos Futuros	p. 81
	Referências	p. 83
	Anexo A – Frequências das Notas Musicais	p. 87
	Anexo B – Lista de Ações Reconhecidas pelo <i>Device Driver</i>	p. 88

Lista de Figuras

1	Remington number 1	p. 16
2	Independência vertical das teclas	p. 17
3	Leiaute do padrão DVORAK	p. 17
4	Ilustração do padrão QWERTY, por Christopher Sholes - 1878	p. 27
5	O signo ideográfico <i>mãe</i> no sistema Bliss.	p. 38
6	Exemplo de matriz 8x8	p. 42
7	O dispositivo e suas camadas	p. 44
8	Protocolo de comunicação serial	p. 45
9	Dois acessos simultâneos no sistema por infravermelho.	p. 47
10	Múltiplo acesso provocado por objeto: o retângulo 3.10-7.14.	p. 47
11	Exemplo de matriz 8x8 com múltiplo acesso	p. 50
12	Exemplo de transmissão serial de 16 bits (matriz 8x8)	p. 51
13	Exemplo de matriz 8x8 reconstruída	p. 51
14	Divisão do <i>device driver</i> em camadas	p. 52
15	Vetor de células modificadas	p. 54
16	Rodízio das matrizes armazenadas em memória volátil	p. 54
17	Filtragem dos dados pelos módulos adicionais	p. 55
18	Exemplo de seleção e ordenação de múltiplo acesso	p. 56
19	Vetor de células modificadas filtrado pelo módulo <i>SOA</i>	p. 58
20	Identificação da primeira e única liberação de célula: 2.11	p. 58
21	Identificação do primeiro acesso: 4.13	p. 58

22	Remoção das células “sombreadas” 4.11 e 7.13	p. 58
23	Estabelecimento da submatriz M de busca	p. 59
24	Identificação do segundo acesso: 7.11	p. 59
25	Estabelecimento da submatriz M de busca	p. 59
26	Matriz de índices MI	p. 61
27	Lista auxiliar de signos LAS	p. 61
28	Vetor aplicado à matriz MI	p. 63
29	Menu principal da aplicação base	p. 67
30	Janela de criação	p. 68
31	Desenho do mapa de signos	p. 71
32	Atribuição dos signos	p. 71
33	Seleção e ordenação dos módulos	p. 73
34	Configuração extra dos módulos	p. 73
35	Armazenar ou descartar um <i>perfil</i>	p. 74
36	Submenu de recuperação de um perfil	p. 75
37	Submenu de ativação de um perfil	p. 76
38	Possibilidades de navegação da aplicação base	p. 76

Lista de Tabelas

- 1 Exemplos de ações remetidas pelo *device driver* ao SO p. 62
- 2 Relação das cores para as tarefas de configuração p. 70
- 3 Frequências das notas musicais de 8 oitavas da escala igualmente temperada. p. 87
- 4 Ações reconhecidas pelo *device driver* e remetidas ao SO p. 88

Resumo

O presente trabalho tem como temática o desenvolvimento de uma tecnologia alternativa à entrada de dados ao computador chamada *COLMÉIA*. Baseada em soluções já conhecidas, como o teclado e o *mouse*, essa nova tecnologia procura expandir as capacidades de interação homem-máquina, utilizando linguagens mais amplas e flexíveis do que as oferecidas pelos meios atuais. Para tanto, permite a personalização (selecção) dos sinais utilizados na linguagem de interação entre homem e computador, possibilitando assim construir conjuntos específicos de sinais para categorias típicas de tarefas ou usuários.

Seu propósito maior é o de servir como uma tecnologia assistiva computacional voltada à educação especial. Além dos aspectos técnicos, outro importante requisito que norteou a pesquisa da *COLMÉIA* foi o seu custo final de produção, dado o panorama educacional brasileiro em termos de parque tecnológico instalado e investimentos em novas tecnologias.

Abstract

This work describes an alternative computer data input called *COLMÉIA*. Based on known solutions, like the keyboard and the *mouse*, this new technology tries to expand human-computer interaction capacities through more flexible languages than that provided by the common devices. This feature is possible because *COLMÉIA* allows the user to select which signs will be used during the human-computer interaction, creating, therefore, specific sets of signs to inumerous kinds of users and tasks.

The *COLMÉIA*'s major target is to serve special education as a computacional assistive technology. Beyond technical features, another importante requirement was the final production cost, given the actual Brazilian educational situation - their installed technological park and investiments in new technologies.

1 Introdução

A informática é, para o mundo contemporâneo, uma poderosa ferramenta de tratamento de dados e de realização de tarefas. Os microcomputadores, a partir dos anos 1980, têm revolucionado o cotidiano das pessoas. Contudo, apesar dos computadores possuírem recursos para operar com uma escala muito grande de signos - e conseqüentemente oferecer um canal de interação bastante amplo - ainda muito pouco desse potencial é utilizado para a representação de dados computacionais. Em acordo com a sua arquitetura interna - baseada na proposta de von Neumann, a entrada de dados é realizada por meio de um conjunto finito de signos que, posteriormente, são processados de acordo com as regras armazenadas na memória do sistema. Nota-se, portanto, a dependência direta entre os dados inseridos no sistema e o processamento global do mesmo.

Um dos motivos mais nítidos para esse conjunto de signos ser restrito foi o reduzido campo inicial de aplicação dos computadores. Desenvolvidos inicialmente para realizar operações matemáticas e, em seguida, para a composição de documentos na língua inglesa, apenas símbolos alfanuméricos e de pontuação de texto eram necessários. Na metade do século XX, com a crescente produção de computadores e dadas as divergentes e pouco amigáveis interfaces vigentes na época, fez-se necessária a criação um padrão de signos a serem usados na interação homem-máquina. Em 1968, a American National Standards Institute (ANSI) criou uma tabela de símbolos a ser utilizada como padrão nos computadores: a American Standard Code for Information Interchange (ASCII), que contém 97 letras, numerais arábicos, símbolos matemáticos, de acentuação e de pontuação para elaboração de documentos na língua inglesa. Nessa tabela, para cada símbolo há um correspondente em número hexadecimal que pode, facilmente, ser compreendido pelo computador - bastando convertê-los para números binários. Estava resolvido, naquele momento, o modo de interação homem-máquina em baixo nível.

A padronização de tabelas como a ASCII tem suas origens ainda mais remotas. Em

1874 a empresa Remington & Sons produziu a primeira máquina de escrever comercial, a Remington Number 1¹ (Figura 1), contendo aproximadamente os mesmos símbolos que posteriormente originaram a tabela ASCII - ambos regidos pelas necessidades (e particularidades) da língua inglesa. Desde então, o arranjo espacial de símbolos e teclas tem sido uma constante tanto nas máquinas de escrever quanto nos teclados de computador. Devido às teclas superiores esquerdas serem as letras Q, W, E, R, T e Y, denomina-se hoje em dia esse padrão como QWERTY². Nos países de línguas distintas da inglesa, foram feitas pequenas adaptações para representar símbolos mais particulares. Por exemplo, no Brasil foi adicionada a tecla ç e trocados alguns sinais de acentuação - passando esse padrão a se chamar ABNT-2. No alemão, o símbolo ö teve também a sua inclusão dentre os símbolos simples (e adicionou-se nos teclados uma tecla correspondente ao símbolo, facilitando o seu uso).



Figura 1: Remington number 1

Apesar dos teclados não possuírem as mesmas limitações mecânicas da máquina de escrever, o arranjo espacial permaneceu o mesmo: traçando qualquer linha vertical sobre o plano do principal conjunto de teclas, onde se encontram os símbolos alfanuméricos, não há a ocorrência de duas teclas que possuem ponto central coincidente, como demonstra a Figura 2.

Além disso, o padrão QWERTY, embora não presente, dentre os vários padrões de teclado, o melhor desempenho em termos de digitação, é o mais disseminado. O padrão DVORAK, por exemplo, apresenta maior eficiência ao reordenar as posições das teclas, a partir de dois princípios diferenciados que nortearam a sua construção: facilidade no uso e

¹REMINGTON PRODUCTS COMPANY. *Remington History*. 2002. Disponível em: <<http://www.remington-products.com/australia/corpinfo/history.html>>.

²Criado por Christopher Sholes, modelo patenteado em 1878.



Figura 2: Independência vertical das teclas

agilidade na digitação. Entretanto, o princípio de evitar o choque das alavancas das teclas - tornando-as independentes verticalmente - foi herdado sem alterações do QWERTY.

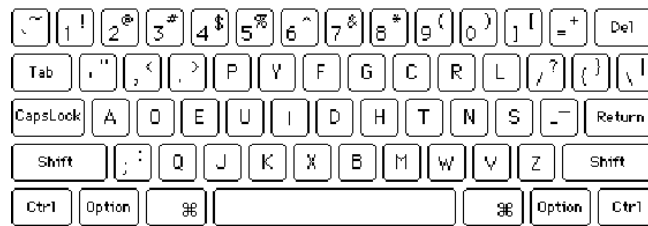


Figura 3: Leiaute do padrão DVORAK

Esse mapa fixo de teclado, tanto na disposição das teclas quanto na ordenação dos símbolos, provê um meio restritivo de entrada de dados: não é possível adicionar teclas ou reorganizá-las de forma a melhorar a sua utilização. O próprio padrão DVORAK, embora considerado um avanço em termos de usabilidade, também possui limitações: não implementa a troca de teclas por outras maiores ou menores (o tamanho mais comum é de 12,7 mm x 12,7 mm na cabeça e 19,1 mm x 19,1 mm entre os seus pontos centrais na base) ou mesmo de cores variadas. Aplicações mais específicas do uso do teclado, como por exemplo na educação especial (BRASIL, 2000), são um caso em que a reorganização ou destaque de certas teclas em muito facilitariam o uso do computador por parte dos aprendizes, permitindo-lhes, entre outras coisas, participar mais ativamente das atividades regulares de ensino.

Mas não é só nas situações específicas que se percebe a necessidade de teclados mais especializados. Em 2002, com a implantação da moeda € (Euro) na Europa, surgiu um problema de limitação dos teclados: eles não possuíam uma tecla correspondente ao símbolo €, dificultando assim o uso do mesmo. A solução adotada para esse problema foi a adição desse símbolo nos novos teclados produzidos a fim de prover um acesso rápido via teclado, além de uma reimplementação do programa interpretador (*device driver*) para aceitar essa mudança. Alterações dessa natureza, apesar de raras, comprovam a rigidez estrutural do periférico.

No caso do *mouse*, que foi criado por Douglas Engelbart³, seu modo de utilização tem princípio oposto ao do teclado: uma interface bastante simples (em média dois botões e um sensor de movimentos em um plano) com a qual é possível acionar diversos eventos/ações escolhidos pelo usuário - que são oferecidos pelo ambiente gráfico, como mover objetos, abrir e salvar documentos, entre outros. Esse avanço, embora tenha proporcionado um considerável salto tecnológico, ainda requer um controle fino dos movimentos que pode, em alguns casos particulares, inviabilizar o seu uso.

Nota-se no caso do *mouse*, em particular, a dependência direta do periférico com o contexto visual em que se encontra (ao contrário do teclado, que provê grande parte dos seus recursos diretamente na sua própria estrutura, o *mouse* se vale de recursos extras para o seu funcionamento). Essa desassociação entre recursos de *hardware* e possibilidades de ação representa uma verdadeira revolução em comparação ao dispositivo anterior, pois o conjunto possível de signos é ilimitado e adaptável em tempo real.

Devido a sua grande flexibilidade, o *mouse* hoje é um dispositivo de entrada de dados tão necessário quanto o teclado - dado o avanço das interfaces gráficas dos programas e dos recursos oferecidos pelos sistemas operacionais. Por isso, o teclado e o *mouse* podem ser considerados como meios convencionais de entrada de dados ao computador.

Embora os dois meios convencionais supram as necessidades mais elementares da interação homem-máquina, a sua utilização excessiva e/ou errônea pode provocar sérios danos à saúde, em especial lesões por esforços repetitivos (LER) e distúrbios osteomusculares relacionados ao trabalho (DORT)⁴. Esse fator pode ser ainda mais nocivo quando há limitações na interação com o computador, onde as causas são outras doenças ou necessidades especiais (deficiência motora, por exemplo) que comprometem - ou mesmo impossibilitam - o uso das ferramentas mais usuais como o teclado ou o *mouse*. Para tanto, tem sido necessário o desenvolvimento de uma série de soluções tecnológicas, alternativas aos meios convencionais, que visam atender a uma demanda específica de tarefas ou usuários portadores de necessidades especiais - tendo como propósito principal compen-

³Histórica e tecnologicamente, cabe aqui ressaltar que, apesar das diversas referências atribuindo a invenção à Xerox PARC - Palo Alto Research Center - manteremos o criador original do dispositivo. Engelbart, do Stanford Research Institute, de fato inventou o *mouse* em 1963. Seu modelo, bastante experimental, utilizava duas rodas dispostas perpendicularmente para a detecção de movimentos. No começo dos anos 70, Bill English (da PARC) aperfeiçoou-o, transformando o *mouse* em um periférico muito semelhante ao que usamos hoje em dia: utiliza-se uma bola ao invés das rodas para identificar os movimentos.

⁴ÁREA TÉCNICA DE SAÚDE DO TRABALHO. *Lesões Por Esforços Repetitivos (LER) / Distúrbios Osteomusculares Relacionados ao Trabalho (DORT)*. Brasília: Ministério da Saúde, 2001. (A - Normas e Manuais Técnicos, v. 103).

sar as limitações inerentes a cada caso. Contudo, essa baixa demanda gera baixa escala de produção e, conseqüentemente, alto custo, trazendo mais um agravante discriminatório a esses usuários.

1.1 Objetivos

O foco principal do estudo reside na proposição e descrição de uma tecnologia de entrada de dados alternativa ao teclado e ao *mouse*, adaptativa e expansível (permite a agregação futura de novos componentes), com vistas à aplicabilidade na área de educação especial (tecnologia assistiva). Além desse objetivo geral, alguns objetivos específicos são previstos no projeto:

1. *Classificação das tecnologias de aquisição de dados*: a fim de levantar o estado da arte das tecnologias de entrada de dados, será realizada uma pesquisa dos métodos de interação com o computador e as suas respectivas implementações, a qual servirá de base de conhecimento para a escolha das tecnologias que irão compor a **COLMÉIA**.
2. *Especificação da tecnologia*: detalhamento de todas as camadas da **COLMÉIA**, abrangendo os seus níveis físico (*hardware*) e lógico (*software*):
 - (a) Dispositivo de aquisição de dados;
 - (b) *Device driver* de controle;
 - (c) Aplicação base de configuração do dispositivo e *device driver* de controle.
3. *Linguagem de interação homem-máquina*: desenvolvimento de uma tecnologia (baseada em eventos⁵ ao invés de caracteres) que abrigará uma linguagem ampla e flexível - para prover uma gama maior de signos, como por exemplo palavras, sons e mesmo ações;
4. *Incorporação de outras funcionalidades*: além das funções básicas, uma série de recursos que pré-processarão os dados de entrada poderão ser incluídos *a posteriori*, a fim de melhorar essa interação junto ao computador e suas diversas aplicações, como por exemplo correção de acessos indevidos ao dispositivo;

⁵Esse termo será melhor definido no Capítulo 5.

5. *Aplicabilidade da tecnologia*: análise de possibilidade da utilização da tecnologia na educação especial, onde enquadram-se portadores de necessidades especiais e não portadores em um mesmo ambiente - suas compensações e potencialidades.

1.2 Justificativa

“Cabe a todos, principalmente aos setores de pesquisa, às Universidades, o desenvolvimento de estudos na busca dos melhores recursos para auxiliar/ampliar a capacidade das pessoas com necessidades educacionais especiais de se comunicar, de se locomover e de participar de maneira cada vez mais autônoma do meio educacional, da vida produtiva e da vida social, exercendo assim, de maneira plena, a sua cidadania.” (BRASIL, 2001)

O estado da arte do referido campo de interesse, a interação homem-máquina voltada às necessidades de tarefas ou usuários especiais e, em particular, à educação especial, ainda é muito restrito e, por esse motivo, têm sido pouco discutidas as pesquisas na área. Os trabalhos desenvolvidos no país, embora significativos e de notório reconhecimento⁶, não conseguem acompanhar o ritmo de crescimento brasileiro (viabilização da implantação do parque tecnológico nas instituições de ensino e diversificação das necessidades especiais atendidas pelas tecnologias desenvolvidas):

- 2002: aumento de 36% na educação inclusiva (BRASIL, 2003);
- 2003: aumento de 30,6%: “a cada 100 alunos com deficiências, 29 estão matriculados em classes comuns”⁷. Isso equivale, em números absolutos, a 144.583 das 358.987 crianças matriculadas nas instituições de ensino do país (BRASIL, 2004).

Além disso, tais estudos não vislumbram, em sua essência primária, uma ferramenta que possua a capacidade de servir tanto para portadores de deficiência como para alunos considerados normais. Como a educação especial prioriza a integração de alunos portadores de deficiência e de alunos não portadores nas classes comuns de ensino, abre-se uma lacuna para o desenvolvimento de alternativas tecnológicas que possam maximizar essa integração (minimizando as limitações ou diferenças entre os alunos). Essa priorização está sustentada por sólida base legal:

⁶Destacam-se, entre outros, os trabalhos de (SACI, 2004), (SANTAROSA, 1994) e (BEZERRA, 2003).

⁷AGÊNCIA ESTADO. *Cresce a inclusão de alunos especiais em escolas regulares*. setembro 2003.

- Constituição Brasileira de 1988 - artigo 208 III;
- Estatuto da Criança e do Adolescente (BRASIL, 1990);
- Declaração de Salamanca (UNESCO, 1994): “as pessoas com necessidades educativas especiais devem ter acesso às escolas comuns que deverão integrá-las numa pedagogia centralizada na criança, capaz de atender a essas necessidades”;
- Lei de Diretrizes e Bases da Educação Nacional (BRASIL, 1996);
- Resolução CNE/CEB nº2 (CNE/CEB, 2001);
- Parâmetros Curriculares Nacionais (BRASIL, 1998);
- Lei Federal nº 7.853 (BRASIL, 1989), que sanciona: “o incentivo à pesquisa e ao desenvolvimento tecnológico em todas as partes as áreas de conhecimento relacionadas com a pessoa portadora de deficiência”.

Todos esses instrumentos legais legitimam a educação especial em detrimento do ensino especializado e excludente que, à primeira vista, pode ser constituído dadas as circunstâncias que envolvem todo o processo educacional dos alunos especiais. São também igualmente unânimes em destacar a integração entre todos os alunos, independentemente de sua condição física, mental ou qualquer outro atributo potencialmente discriminatório. Apesar disso, alguns entraves à essa condição tão natural de igualdade podem surgir quando confrontado o direito básico do portador de necessidades especiais com o seu cotidiano.

Contudo, sob uma óptica técnica, o leque de possibilidades é bastante amplo e promissor, como destacam as novas tecnologias de interação homem-máquina, tais como: teclado e *mouse* integrados⁸, *touchpads* multiponto sensíveis à pressão⁹ e teclados projetados¹⁰. Dentro dessa nova perspectiva de oportunidades, abrem-se duas claras vertentes de uso:

1. *Uso complementar*: além do teclado e do *mouse*, vias comuns de interação com a máquina, é possível adicionar outras funcionalidades a partir do periférico complementar. Dessa forma, cria-se ao usuário uma terceira via de interação, porém

⁸FINGERWORKS, INC. *FingerWorks*. 2003. Disponível em: <<http://www.fingerworks.com>>.

⁹CHEN, T. T. H.; FELLS, S.; MIN, S. S. Flowfield and beyond: applying pressure-sensitive multi-point touchpad interaction. In: . Baltimore, EUA: IEEE International Conference on Multimedia & Expo, 2003.

¹⁰CANESTA. *Canesta Keyboard*. 2002. Disponível em: <<http://www.canesta.com/products.htm>>.

adaptável às mais diferentes situações ou aplicações (teclas de atalho, combinações, etc.). Seu uso será, portanto, bastante reduzido e/ou específico;

2. *Uso alternativo*: ao invés de adicionar uma terceira via de interação, alterna-se do uso combinado teclado+*mouse* para a tecnologia alternativa. Além de suprir todas as necessidades desses dois anteriores, ainda permite a incorporação de novas funcionalidades - desde que suportado pelos seus recursos físicos e lógicos.

A primeira vertente, apesar de possível, é desaconselhada pelo fato de não resolver a problemática instalada. Ao contrário dessa, a segunda vertente insere o usuário em um ambiente passível de modificações sensoriais e “comportamentais” e, portanto, em um ambiente interativo mais rico. Tais possibilidades ampliam as capacidades de uso do computador, melhorando em muito a sua condição de instrumento de apoio em casos mais complexos, como por exemplo a educação especial.

Em relação às tecnologias de aquisição de dados, tem-se construído componentes eletroeletrônicos cada vez mais baratos e funcionais, o que, somado ao fato anterior, só vem contribuir para a viabilidade dessas soluções alternativas. Apesar dessa queda nítida dos preços, no Brasil esse quadro poderia ser bem melhor: o país é o maior produtor mundial de silício e detentor das maiores reservas desse minério, mas paradoxalmente não possui sequer uma indústria de produção de processadores. Ou seja, todo o minério é obrigatoriamente exportado para voltar ao país de origem beneficiado na forma desses componentes eletrônicos - com elevado valor agregado.

Atualmente, as ferramentas de interação homem-máquina de característica adaptativa são, para os padrões brasileiros, demasiadamente caras ou tecnologicamente insuficientes para a sua ampla implantação em instituições de ensino. Teclados de conceitos como o Intellikeys (INTELLITOOLS, 2003)¹¹, por exemplo, são produtos importados e, por esse e outros motivos, chegam ao mercado com um custo final bastante elevado (cotação de julho de 2004 em (CLIK, 2002): R\$2.160,00). Para se ter uma idéia, uma sala de aula com 12 computadores¹² teria um custo adicional de R\$25.920,00 com a inclusão desse produto na sua infraestrutura - um valor hoje proibitivo para as escolas públicas, tendo em vista as condições financeiras de tais instituições.

¹¹Utilizados, por exemplo, pela Organização Nacional dos Cegos da Espanha - ONCE.

¹²Quantidade de computadores recomendada para uma sala de aula do tipo A conforme (BRASIL, 2000, anexo III).

1.3 Hipóteses

Busca-se, portanto, uma tecnologia de entrada de dados cuja interface seja passível de modificações para adaptar-se às necessidades de tarefas e usuário especiais, visando com isso amenizar os problemas decorrentes das limitações impostas pelas circunstâncias que envolvem esses usuários.

É sabido que as várias deficiências e seus graus de comprometimento (CAMPOS; SILVEIRA, 1998) geram uma demanda muito restritiva de necessidades, tornando quase impossível a aplicação de uma mesma tecnologia para todas. Dessa forma, para delimitar o escopo deste trabalho, é necessária a definição prévia de um público-alvo mais específico. Para os propósitos deste trabalho, os portadores de deficiências físicas, em seus diferentes níveis de comprometimento motor dos membros superiores, devem ser os beneficiados mais diretos dos resultados aqui conseguidos.

Apesar de buscar a criação de um novo padrão para a interação homem-máquina, essa tecnologia será fundamentada nos atuais meios convencionais - teclado e *mouse* - visto que atualmente tais componentes constituem um padrão *de facto* de entrada de dados ao computador. Entretanto, não será foco deste trabalho produzir um meio complementar na interação homem-máquina, mas sim buscar o aperfeiçoamento tecnológico dos meios convencionais para maior flexibilidade ao usuário e servir como um meio alternativo de entrada de dados de baixo custo.

1.4 Resultados Esperados

Espera-se, ao final do trabalho, além dos objetivos já mencionados, uma ferramenta que possa, de fato, atuar nos diversos segmentos educacionais, sem com isso acarretar em perda de desempenho nas situações mais rotineiras, como por exemplo a digitação de um documento. Uma vez que o leque de opções é explorado para adequar a ferramenta aos vários campos do saber (ao invés do processo inverso de adaptar o usuário à ferramenta), pode-se utilizar o computador como auxiliar do processo de ensino-aprendizagem. O computador deixa de ser o ator principal, a tecnologia como um fim em si, e passa a ser coadjuvante do processo. Além disso, espera-se que a tecnologia possa ser um diferencial dentro de um processo de ensino-aprendizagem de médio ou longo prazo: operações matemáticas, letras do alfabeto, etc. À medida que o aluno vai aprendendo, vai assimilando

as novas informações num contexto cada vez mais rico e construtivista, e esse novo conhecimento vai sendo incorporado ao periférico com novos “botões” ou funcionalidades - de acordo com o ritmo e vontades de cada aluno.

Já no tocante direto às necessidades especiais, espera-se que a tecnologia poderá não somente compensar as limitações, mas também potencializar as capacidades de cada indivíduo. Uma vez estabilizadas as relações entre os usuários no uso do computador, que cada um possa, através desses mesmos recursos, destacar-se por suas qualidades e capacidades, e não por sua condição física ou mental. Essa é a contribuição esperada para inclusão digital dos considerados excluídos devido à sua posição marginal no processo tradicional de ensino e, conseguinte, de mercado de trabalho.

E, por último, espera-se que o produto derivado dessa pesquisa possa ser disponibilizado com um custo mínimo de produção, no anseio de chegar ao usuário final com um baixo custo de aquisição, tendo em vista o potencial público-alvo em âmbito nacional.

1.5 Metodologia

Para a criação de uma tecnologia baseada em soluções de entrada de dados já existentes, primeiramente é necessário analisar os recursos disponibilizados pelo teclado e pelo *mouse* como instrumentos de interação homem-máquina. Há muitas formas de expandir o conjunto de signos inicialmente proposto pelo teclado QWERTY, como por exemplo através de acesso seqüencial (~ e a para formar ã) ou composto (Shift + a para compor A) de teclas. Em relação ao *mouse*, seus eventos podem ser estendidos com associações entre os botões e as direções (movimentos). Com o devido rastreamento das possibilidades de combinação, é possível delinear os tipos de signos para o novo dispositivo (caracteres, sons, etc.).

Outro passo importante será pesquisar as tecnologias atualmente utilizadas em dispositivos não convencionais de interação homem-máquina, como por exemplo: sistemas resistivos e capacitivos; membrana de contato; raios infravermelho, etc. Com base nessa pesquisa, é possível escolher uma ou mais tecnologias para compor o periférico e, posteriormente, a sua granularidade (quantidade de botões ou pontos de acesso por uma dada área delimitada). A quantidade de pontos de acesso deve ser um valor aceitável e coerente com as normas especificadas para a(s) tecnologia(s) escolhida(s). Como consequência desse valor, definem-se as dimensões do dispositivo, a área de acionamento das teclas e

os materiais envolvidos na sua confecção. Finalizada a etapa de seleção de tecnologias, proteja-se um modelo eletrônico e, em seguida, um protótipo.

Na próxima etapa, projetar-se-á o *device driver* de controle do *hardware*. Para a implementação, serão seguidas duas recomendações:

1. Programação modular (base + módulos);
2. Código em alto nível, portátil e aberto (livre acesso ao código fonte).

Uma etapa preliminar de testes de integração *hardware-device driver* é realizada para, então, desenvolver o *software* de configuração da **COLMÉIA**. O *software* possuirá interface para a criação e alteração de n configurações ou *perfis*.

E, por último, faz-se a integração dos três componentes: periférico, *device driver* e *software*, formando uma tecnologia única de entrada de dados programável.

2 *Os Meios de Entrada de Dados*

A manutenção da similaridade entre os meios de entrada de dados, atuais e futuros, é importante para o desenvolvimento e uso de novas tecnologias, visto que cada nova solução sugere uma (re)adaptação do usuário para bem utilizá-la.

Tal similaridade, embora muitas vezes limitante, procura minimizar os custos de uma mudança tecnológica, permitindo assim (re)utilizar o conhecimento prévio do uso de um periférico semelhante e que já pode ser aplicado, objetivando assim sua maior aceitação. Partindo desse princípio de compatibilidade, serão utilizados os meios convencionais de entrada de dados, teclado e *mouse*, como instrumentos de referência para a descrição de um meio alternativo de interação. Com isso, espera-se reduzir ao máximo os obstáculos de adaptação e uso do novo periférico.

Todavia, além das tecnologias convencionais de aquisição de dados, outras têm sido utilizadas para compor soluções voltadas à interação homem-máquina, seja com o intuito de complementar ou mesmo substituir os atuais periféricos. Essas soluções, desenvolvidas para fins mais específicos, têm utilizado recursos diferenciados e inovadores para prover um melhor acesso ao computador. Devido a isso, serão incluídos nessa pesquisa os meios de entrada de dados que apresentam inovação ou aperfeiçoamento tecnológico em relação aos meios convencionais.

Será apresentado, ao longo do capítulo, um estudo crítico dos atuais meios convencionais e alternativos de entrada de dados. Tal estudo deve, posteriormente, auxiliar na definição das tecnologias que irão compor a *COLMÉIA*.

2.1 **Os Meios Convencionais de Entrada de Dados**

Para uma compreensão mais detalhada de uma tecnologia de aquisição de dados, é importante conhecer o seu modo de uso (por parte do usuário) e de funcionamento

(integração com o computador e o sistema operacional). O teclado e o *mouse*, pela sua difusão como instrumentos de entrada de dados, terão omitida a sua explicação de modo de uso. Entretanto, será apresentado a seguir o modo de funcionamento de ambos para um entendimento melhor das tecnologias envolvidas.

2.1.1 O Teclado

O teclado, como já mencionado, possui atualmente diversos padrões e leiautes. Para o nosso estudo, usaremos o modelo que originou os demais, o padrão US-ASCII (americano) - criado diretamente a partir da tabela ASCII e do arranjo das teclas da máquina de escrever. O seu padrão pode ser visualizado na Figura 4.

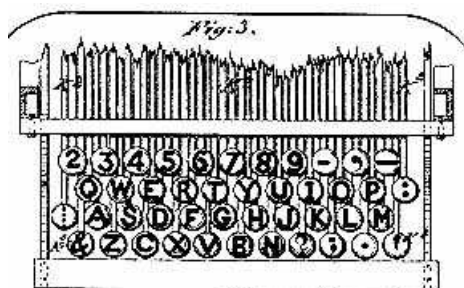


Figura 4: Ilustração do padrão QWERTY, por Christopher Sholes - 1878

Para uma melhor compreensão do modo de funcionamento de um teclado, dividiremos as suas teclas em quatro grupos ou tipos (modo de divisão baseado em (SCHMIDT, 1995)):

1. *Normais*: são interpretadas como remetentes diretas de símbolos ao sistema. Enquadram-se nesse grupo as teclas de números, letras e símbolos matemáticos, onde o símbolo possui uma referência visual no teclado;
2. *De mudança de estado*: provocam apenas alteração dos símbolos correntes nas teclas de um dado subconjunto, como é o caso da tecla Caps Lock que altera o valor das letras - trocando a caixa baixa pela caixa alta e vice-versa;
3. *Compositivas*: operam em conjunção com as teclas normais, como é o caso de Shift, Ctrl, Alt e Altgr. Sua função principal é permitir uma gama maior de símbolos utilizando as teclas *normais* em combinação com as compositivas, e somente em casos muito especiais quando acionadas isoladamente geram uma reação

visível do sistema. Por exemplo, tomemos Shift, que em conjunção com a forma A, mas Shift isoladamente não provoca uma reação visível;

4. *Mistas*: variam de acordo com a linguagem corrente do sistema operacional. Nota-se aqui a interdependência entre a *hardware* e a *software*: dependendo da linguagem e suas características, certas teclas podem operar como *normais* ou *compositivas*. Exemplo: no caso da língua inglesa, a tecla contendo os símbolos ~ e ` ativarão diretamente tais símbolos, enquanto que na língua portuguesa haverá uma espera de uma nova entrada (vogal ou consoante) para enviar o símbolo correto: combinado ~ com a forma-se ã, enquanto que ~ com q o resultado é nulo (descarta-se o símbolo por não fazer parte da língua). (TANENBAUM, 1992)

Além da configuração das teclas, é possível também ajustar:

- *Taxa de repetição (rate)*: mantendo uma tecla pressionada, esta pode transmitir várias vezes o símbolo escolhido. Geralmente esse valor fica em torno de 30 cps (caracteres por segundo), mas esse valor é passível de ser alterado (a escala possível de valores é de 2 a 30 cps). Esses valores são para a arquitetura Intel. Para a arquitetura SPARC, a título de comparação, esses valores são de 0 a 50 cps (com média de 20 cps).
- *Tempo de espera (delay)*: quantidade de tempo, em milissegundos, antes de começar a repetir o símbolo. Em média, esse valor é de 250 ms, mas pode chegar até 1000 ms. Nos computadores SPARC a variação fica entre 10 e 1440 ms (média de 200 ms).

2.1.2 O Mouse

Quanto ao *mouse*, será usado como referência o periférico de dois botões, padrão dos computadores pessoais do tipo PC (*Personal Computer*), o caso mais comum. Há variações desse modelo, com três botões, botão com roda (*wheel mouse*), entre outros, contudo será utilizado aqui o mesmo critério: o modelo originalmente desenvolvido será considerado para o estudo.

O *mouse* possui dois submodos de funcionamento: botões e direcional. Os botões são muito semelhantes às teclas mistas do teclado, onde cada tecla ora corresponde a um símbolo específico, ora é utilizado em combinação com outras teclas. A necessidade

de teclas mistas ocorre porque alguns sistemas operacionais utilizam três botões (como o GNU/Linux, por exemplo), e para resolver esse problema, são usados os dois botões simultaneamente. Ou quase simultaneamente, pois as restrições da comunicação serial impossibilitam a real simultaneidade de teclas. Na prática, as teclas são acionadas seqüencialmente em um prazo muito curto de tempo, em média 100 ms (o prazo estabelecido é ajustável em nível de *software* - no sistema operacional). Se esse tempo para ocorrer a combinação dos botões se esgotar, o sistema entende a ação como duas teclas acionadas distintamente.

Quanto ao modo direcional, o *mouse* possui três eixos:

$$\left. \begin{array}{l} \text{Eixo } x \\ \text{Eixo } y \end{array} \right\} = \text{Plano da tela}$$

$$\left. \begin{array}{l} \text{Eixo } z \end{array} \right\} = \text{Barra de rolagem}$$

Os eixos x e y referem-se às direções possíveis do cursor na tela: um plano onde cada ponto pertencente a esse espaço pode ser representado por 2 vetores: x e y ; e o eixo z tem a mesma função da barra de rolagem. Apesar de não pertencer ao modelo seguido de *mouse* com apenas dois botões, o terceiro eixo será relevado e considerado nessa pesquisa tendo em vista as suas características e potencialidades de acessibilidade.

Entre os eixos x , y e z a combinação se dá apenas sobre os dois primeiros eixos (altura e largura do plano da tela); o eixo z , por implementar a rolagem de textos ao longo de uma área de leitura, não possui combinações com os outros dois eixos. Já em relação aos botões, esses se combinam entre si, com os eixos x e y e as teclas do teclado convencional, criando um meio mais rico de recursos para a interação homem-máquina em ambientes gráficos.

2.2 Os Meios Alternativos de Entrada de Dados

Os meios alternativos, pesquisados e selecionados, apresentam tecnologias em *hardware* e *software* de aspecto inovador e adaptadas a diversas necessidades e requisitos (CAMPOS; SILVEIRA, 1998). Para uma melhor compreensão do propósito e das funcionalidades de cada solução pesquisada, foi criada uma classificação baseada nos seguintes critérios (em ordem decrescente de prioridade):

- *Método de ativação*: forma de acionamento dos signos baseada nos recursos físicos (*hardware*) e lógicos (*software*) utilizados na implementação;
- *Funções disponíveis*: as ferramentas e facilidades que a solução oferece ao usuário para a interação homem-máquina;
- *Diferencial*: ferramenta ou facilidade identificada em uma solução que mereça destaque pela inovação.

Dos três critérios supra descritos, o primeiro é o único que limita o uso do periférico por parte do usuário; os demais atuam como agentes compensatórios ou potencializadores. Devido a isso, o *método de ativação* será tratado como o critério principal de classificação, secundado pelos demais que destacam as particularidades de uma solução em especial. Cabe aqui ressaltar que os critérios foram definidos a partir da pesquisa acerca das tecnologias de aquisição de dados ao computador.

A seguir, a definição de cada categoria de tecnologias organizada segundo o critério método de ativação.

2.2.1 Ativação por Toque

As soluções de ativação por toque são compostas por um dispositivo físico, meio de acesso aos signos utilizados na interação, e um *device driver* de controle. O dispositivo é implementado com tecnologias que utilizam o contato entre o usuário e o periférico, seja esse direto ou indireto, como forma de acionamento dos signos - provendo retorno tátil e, em alguns casos, sonoro.

Mapa de Teclas

O contato entre o usuário e o computador se dá por meio de teclas ou botões, possibilitando assim a intermediação do acesso por qualquer objeto (caneta, etc.). Entretanto, devido ao processo mecânico de acionamento, o uso de força em dispositivos dessa natureza pode comprometer a sua utilização em se tratando de pessoas portadoras de necessidades especiais. Como exemplo de mapa de teclas, além das diversas variações do teclado convencional (mapa de signos alterado para fins específicos) temos: pedal e ativador binário.

Sistema Resistivo

O sistema resistivo, em relação ao seu dispositivo, consiste em três camadas: uma inferior condutiva, uma intermediária dotada de corrente elétrica, e uma superior metálica resistiva. Ao toque do usuário, a pressão resultante gera um contato entre as camadas externas em um determinado ponto, o qual é caracterizado como ponto de toque. Esse ponto, célula de uma matriz, é enviado ao computador que, em seguida, relaciona o ponto com o signo escolhido.

Por sobre as três camadas, há ainda uma película protetora, resistente a arranhões. Dessa forma, é possível usar o sistema resistivo em equipamentos móveis. Apesar de ter um custo relativamente baixo (comparado às tecnologias semelhantes em termos de uso), pode ser facilmente danificado por objetos pontiagudos ou cortantes. Um exemplo comum é a tela sensível ao toque baseada em sistema resistivo, muito utilizada em *palmtops*. Outro problema comum nesse sistema é decorrente do mau uso do apontador: projetado para ser utilizado em um ângulo normal ao plano da película, pode reproduzir um resultado diferente com ângulos fora desse centro (por exemplo, ao usar o apontador em um ângulo próximo de 30°, poderá ser traçada uma semi-reta ao invés de marcar apenas um ponto). Em consequência dessa restrição, é muito difícil o múltiplo acesso ao dispositivo (vários toques/acessos simultâneos), por causa das interferências mútuas dos apontadores (A película superior sofre distorções indesejadas da sua forma).

Sistema Capacitivo

No sistema capacitivo, há apenas uma camada dotada de uma matriz carregada eletricamente. Ao toque do usuário, a energia que se encontrava na matriz é transferida para o corpo humano, diminuindo a carga no sistema. Esse decréscimo é identificado pelos sensores localizado nas bordas do sistema, e o ponto de toque é transmitido ao computador como o signo escolhido.

A vantagem do sistema capacitivo, em relação ao resistivo na confecção de telas sensíveis ao toque, é que o primeiro permite uma transparência visual de até 90% para a visualização da tela disposta atrás do dispositivo, enquanto que o segundo alcança valores próximos de 75%, o que pode representar um sério empecilho para deficientes visuais. Como desvantagens, possui um custo maior de produção, além de tornar obrigatório o contato direto com uma parte do corpo humano ou de um objeto condutivo. Os exemplos

mais conhecidos são: mesa digitalizadora e *tablet* e a tela sensível ao toque baseada em sistema capacitivo.

Sistema por Ondas Acústicas

Em uma superfície plana, são emitidas ondas acústicas (*surface acoustic wave*) em formação de matriz, onde o método de ativação é bastante semelhante ao resistivo, exceto que objetos pontiagudos (uma caneta, por exemplo) podem não funcionar corretamente como apontadores. A vantagem em relação ao próprio sistema resistivo e o capacitivo é que não há uma malha condutora de eletricidade, possibilitando uma transparência do sistema em 100%, beneficiando, em muito, o uso telas sensíveis ao toque baseadas nesse sistema. Em contrapartida, o seu custo bastante elevado de produção em relação a esses mesmos sistemas.

2.2.2 Ativação por Obstrução

O acionamento dos signos se dá pela obstrução dos raios emitidos pelo dispositivo, em formação de matriz, como método de detecção das ações do usuário. Diferentemente do método anterior, não requer qualquer tipo de contato com o periférico.

Sistema por Infravermelho

Apesar de muito semelhante aos sistemas anteriormente citados, em especial o de ondas por alta frequência, aqui a independência do toque permite o uso desse sistema em espaços variáveis, como por exemplo em uma tela sensível ao toque¹ ou mesmo um carro de brinquedo que desvia dos obstáculos (LEGO.COM, 2003).

O sistema por infravermelho, como meio de entrada de dados, consiste em uma malha de raios disposta em um plano. Esses raios são gerados por um lado por LEDs (*Light Emitting Diodes*) e recebidos pelo outro por fotorreceptores (também diodos). Apesar de poderem ser usados em grandes áreas pela boa orientação dos raios, eles são facilmente refletidos em objetos sólidos (paredes, teto, etc.), o que dificulta o seu uso - mesmo em curtas distâncias. Como exemplo, a tela sensível ao toque baseada nesse sistema e

¹A expressão *touchscreen* e a sua tradução mais comum, *tela sensível ao toque*, vão de encontro à classificação aqui proposta, pois existem telas sensíveis ao toque em que não é obrigatório o contato com o periférico. Entretanto, o termo será utilizado na sua definição original para facilitar a compreensão do texto.

teclados projetados (CANESTA, 2002).

2.2.3 Ativação por Movimentos

Nesse método de ativação, os movimentos do usuário definem os signos enviados ao computador. Diferencia-se dos anteriores por basear-se nos movimentos em si, e não apenas na posição final (toque de um botão, obstrução de um raio, etc.).

Mãos

A alteração na posição das mãos e dos dedos (relativa e absoluta em relação à mão ou ao antebraço) determina os signos acessados. As luvas usadas em realidade virtual e o próprio *mouse*² são um exemplo desse método.

Cabeça

Os movimentos da cabeça, apesar de curtos se comparados aos membros do corpo, também podem ser utilizados como método de ativação. Nesse caso, é provável o uso de recursos extras para o acesso aos signos. O exemplo mais comum é o uso de um emissor *laser* ou infravermelho instalado em um capacete onde, ao movimentar a cabeça, o raio incide sobre uma matriz de fotorreceptores que, dessa maneira, podem discriminar o signo desejado. Outro exemplo são os capacetes usados em realidade virtual.

2.2.4 Ativação Indireta

Os métodos de ativação indireta são aplicativos que interpretam sinais adquiridos por um meio de entrada de dados e já processados pelo Sistema Operacional (SO), atuando como elemento complementar na interação homem-máquina.

Primeiramente, os dados são adquiridos por um ou mais meios de entrada de dados, sem restrição de tecnologia. Esses periféricos não são responsáveis diretamente pelo envio dos signos, mas sim operam apenas como um canal de entrada de valores, ou conjunto de valores, brutos ao computador. Os dados são enviados ao SO que, após um pré-processamento das informações, executa o aplicativo que os converte em signos

²Apesar de poder ser considerado um dispositivo misto, de teclas e movimentos, prevalecerá a sua função prioritária de ativação por movimentos.

válidos na interação homem-máquina. A ativação é dita indireta pois uma ação gerada pelo usuário em um meio qualquer de entrada de dados é processada pelo SO conjuntamente com o aplicativo, podendo assim sofrer alteração no seu conteúdo dependendo da sua configuração, tais como idioma, teclas de atalho, etc.

Reconhecimento de Imagem

Os métodos de ativação indireta são divididos em duas etapas: aquisição e conversão. No caso do reconhecimento de imagem, um dispositivo de aquisição dos dados, uma filmadora por exemplo, capta uma representação gráfica sem qualquer tipo de análise da informação recebida. Essa representação é enviada a um programa que, ao conflitar as imagens captadas (ou parte delas) com informações armazenadas na memória do computador (configuração do sistema e outras imagens já captadas para fins de comparação, pode gerar como saída desse processo um ou mais signos. As vantagens desse processo é que vários signos podem ser reconhecidos em uma mesma imagem, além de permitir a troca dinâmica da base de dados de imagens residentes em memória (troca de alfabetos japoneses *kanji* ou representações pictóricas, por exemplo).

Um ponto que deve ser destacado é que essa interação é dita passiva. Os dados são enviados ao computador sem uma ação direta do usuário, como apertar uma tecla ou obstruir um raio de luz. Como exemplo de reconhecimento de imagem, o aplicativo Dasher (PSL-BR, 2004), que analisa imagens do olho humano para identificar, em tempo real, qual o foco visual do usuário e, assim, determinar o lado do monitor que está sendo observado (cada um dos lados - esquerdo e direito - possui um significado e uma ação específicos).

Reconhecimento de Voz

De todos os meios de entrada de dados, esse é considerado o mais natural ao homem. A voz, captada por um microfone, é enviada a um programa que analisa as diversas informações de uma voz (tom, amplitude e frequência) para retornar os signos relacionados. Entretanto, o reconhecimento de voz tem sido ainda motivo para muitas pesquisas, pois a voz humana possui diversas alterações provocadas por motivos alheios como humor, problemas de saúde, etc. Além disso, a captação do ambiente como um todo deve ser devidamente tratada e filtrada para catalisar apenas a voz isolada, o que tem movido grande parte dessas pesquisas devido a sua complexidade em atingir um nível satisfatório

de qualidade. Hoje em dia, utilizam-se recursos de inteligência artificial para “aprender” a reconhecer uma voz em particular para, em seguida, reconhecê-la como canal válido de entrada de informações. Uma alternativa que tem sido usada, mais simples, é o sopro ao invés do tratamento completo da voz. Como exemplo, o ViaVoice (IBM).

Teclados Virtuais

Os teclados virtuais são compostos por uma interface visual (geralmente uma janela gráfica), onde estão inseridas figuras dos signos, e um ou mais dispositivos de apontamento. Ao utilizar pelo menos um desses periféricos, é possível selecionar uma figura representativa na tela correspondente ao signo de saída.

A grande vantagem dos teclados virtuais é que, pela sua implementação em nível puramente lógico (*software*), não há qualquer limitação física. É possível, portanto, modificar a interface visual, bem como o conjunto de signos disponíveis, em tempo real. As limitações físicas são transferidas aos dispositivos de acesso, que podem ser alternados ou mesmo combinados, formando uma solução expansível. Como exemplo de teclados virtuais, vale destacar: (GONÇALVES et al., 2002), (SACI, 2004) e (LAGARES, 2004).

2.3 Classificação: Análise e Síntese

Ao final deste capítulo, podemos concluir que a classificação foi benéfica para esta pesquisa e válida para dois fins em especial:

1. *Análise*: o processo de classificação é, em essência, o levantamento das semelhanças e diferenças entre os elementos - objeto da pesquisa - e sua sistematização em uma estrutura organizada. Esse processo de comparação, que utiliza critérios de classificação, traça uma visão crítica sobre as soluções tecnológicas (capacidades e limitações);
2. *Síntese*: uma vez classificados as soluções, é possível identificar rapidamente as características de cada uma das mesmas de acordo com a sua posição relativa na estrutura global.

Para a próxima etapa, serão explorados esses dois aspectos na definição das tecnologias de aquisição de dados da **COLMÉIA**.

3 *Os Requisitos para a Escolha das Tecnologias*

No capítulo anterior, foi apresentada uma classificação dos meios de entrada de dados, com relação às tecnologias utilizadas na sua confecção. Nessa classificação, o critério *método de ativação* foi usado para compor as categorias de classificação, os outros dois, *funções disponíveis* e *diferencial*, foram utilizados, de uma forma indireta para a definição das subcategorias, para trazer subsídios para uma melhor compreensão de cada uma das soluções pesquisadas.

Nesse capítulo, será melhor definido o escopo e propósitos do projeto, a partir da definição de cenários de uso e de aplicação. Feito isso, será possível escolher, dentre as categorias e subcategorias, uma ou mais tecnologias que irão compor a *COLMÉIA*.

3.1 Os Requisitos Básicos

Basicamente, a *COLMÉIA* deverá possuir alguns recursos, comumente encontrados em outras soluções, para prover o mínimo de acessibilidade no uso dessa nova tecnologia, mantendo a conformidade com as já utilizadas de modo a reduzir o esforço necessário para a sua aprendizagem. Esses requisitos básicos serão, portanto, a base de formação da tecnologia. Outros requisitos, que abrangem novos recursos específicos ao que se propõe, serão apresentados e discutidos em seguida.

1. *Retorno (Feedback)*: um dos requisitos fundamentais em relação ao acesso ao dispositivo é o retorno que o mesmo provê. A maneira mais comum de retorno é a tátil - as outras modalidades são a visual e a sonora. No caso da *COLMÉIA*, onde vislumbra-se o seu uso para fins educacionais (regular e especial), oferece-se através do dispositivo a primeira (tátil) e, conjuntamente com recursos do compu-

tador (caixas de som e monitor), as outras duas formas, podendo combiná-las sem restrições;

2. *Custo de produção*: outro requisito básico é o custo de produção do dispositivo, cujo número pode representar um entrave à sua confecção em escala;
3. *Força*: a força empregada para acessar o dispositivo, em alguns casos de deficiência física e mental, pode comprometer, quando não proibir, a sua plena interação com o computador.

3.2 As Funcionalidades Adicionais

Além dos requisitos básicos esperados na *COLMÉIA*, alguns recursos extras serão considerados a fim de se prover um verdadeiro ambiente de interação homem-máquina, intuitivo e que permite múltiplas combinações entre os signos.

A seguir, as funcionalidades adicionais.

1. *Alteração do mapa de signos*: a *COLMÉIA*, cujo propósito principal é oferecer uma tecnologia alternativa e programável de entrada de dados, será fundamentada em tecnologias que não limitam ou restringem o seu uso através de interfaces fixas. Dessa forma, serão escolhidas uma ou mais tecnologias que permitem a alteração dinâmica do mapa de signos usado na interação homem-máquina.
2. *Signo composto*: uma vantagem encontrada em alguns dos periféricos pesquisados é a possibilidade de signo composto - uso simultâneo e combinado - dos signos. Esse tipo de múltiplo acesso amplia, em muito, as possibilidades do dispositivo. Por esse motivo, será incorporado à *COLMÉIA*.
Cabe aqui ressaltar que a transmissão serial dos dados ao computador não compromete o uso de signo composto, pois a combinação se dá ao nível do *device driver* (*software*) ou mesmo da placa controladora (*hardware*) do periférico, como pode-se perceber no teclado convencional, que combina as teclas pressionadas mesmo baseando-se em uma comunicação serial (bidirecional) com o computador.
3. *Tipos variados de signos*: além do signo composto, é interessante a inclusão de várias possibilidades para representar os signos usados na interação homem-máquina: ícones, ideogramas, etc. Dessa forma, a *COLMÉIA* permitirá a composição

de uma mapa de signos com quaisquer elementos representativos, podendo, no caso de figuras, variar as suas cores, formatos e tamanhos.

4. *Signos simples e encadeados*: convencionalmente, a forma de acesso aos signos pode ser entendida como uma função injetora, onde cada acesso está relacionado a um signo. Entretanto, não está previsto o acesso um-para-muitos, onde cada acesso implica em vários signos organizados em seqüência. Aliado à funcionalidade anterior, permite que figuras representativas resultem em conjuntos de signos encadeados em seqüência: sílabas, palavras ou mesmo sentenças mais complexas como frases. Um exemplo prático dessa funcionalidade é o *Bliss*, sistema onde cada elemento visual representa a idéia de uma coisa (ideográfico), conforme a Figura 5 (CLIK, 2002).

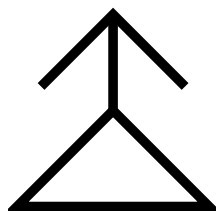


Figura 5: O signo ideográfico *mãe* no sistema Bliss.

O caso inverso, muitos-para-um, onde uma seqüência de acessos concatenados pode representar um único signo, também será considerada como uma funcionalidade útil para a *COLMÉIA*, como por exemplo acionar os símbolos dispostos seqüencialmente em uma linha para realizar a ação “desligar o computador”.

3.3 A Tecnologia Base

Vistas as tecnologias, bem como a sua classificação, e as funcionalidades desejadas para o dispositivo, foi escolhido o método de ativação por sistema por infravermelho. Além de suas características já mencionadas no Capítulo 2, essa tecnologia atende a cinco dos sete requisitos:

1. *Custo de produção*: embora seja uma tecnologia mais cara que a ativação por toque, o custo dos componentes eletroeletrônicos e de transmissão dos dados utilizados é considerado razoável para os padrões do Brasil em termos de produção em escala. Além disso, sua baixa definição, prevista em 600 células por matriz, garante um

tamanho médio de célula (20 mm x 20 mm) para uso semelhante ao de um teclado, e não requer uma grande quantidade de transmissores/receptores de infravermelho - a parte mais onerosa do periférico;

2. *Força*: por não impor qualquer restrição em termos de força empregada no acesso ao dispositivo, o sistema por infravermelho pode ser amplamente utilizado para os mais diversos fins, além de permitir várias alternativas para retorno (*feedback*);
3. *Alteração do mapa de signos*: no sistema por infravermelho não há uma relação direta entre os signos e o ponto de acesso (célula da matriz obstruída pelo usuário). Assim, é possível implementar diversos mapas de signos, obrigando apenas relacionar, previamente, cada célula, ou conjunto de células, com o seu signo correspondente;
4. *Tipos variados de signos*: esse sistema é bastante semelhante a uma tela sensível ao toque, exceto que o plano de fundo não é um monitor, mas sim um plano coberto por uma figura (no caso uma folha A4). Cada área da folha, ou célula, poderá conter qualquer figura, inclusive podendo combinar várias células com uma única figura;
5. *Signos simples e encadeados*: Ao nível do *device driver*, assim como no teclado que recebe um *scancode*¹ (BROUWER, 2002), o número da célula é recebido e, com base em uma tabela que relaciona células e seus valores, repassa ao Sistema Operacional (SO) a sentença correspondente. A sentença, que pode conter *n* caracteres, pode ainda ser pré-processada pelo próprio *device driver*. Por exemplo: a sentença som 440 1, referente à célula 23F, ao invés de enviar ao SO essa expressão, vai gerar um som no auto-falante (*speaker*) do computador - mais especificamente uma nota **lá** no intervalo de 1 (um) segundo. Tem-se, portanto, o conceito de evento: uma sentença que pode significar um ou mais signos (requisito de *signos simples e encadeados*), como a letra a ou a palavra amor, ou mesmo uma ação a ser executada pelo SO. Dessa forma, o *device driver* possui total controle sobre o *hardware*, podendo vir inclusive a corrigir falhas acidentais no acesso, como tremor nas mãos, por exemplo.

¹Código enviado pelo teclado ao Sistema Operacional referente à tecla pressionada.

3.4 As Tecnologias Complementares

Além dos cinco requisitos satisfeitos, o sistema por infravermelho foi escolhido também por permitir que outras tecnologias, aliadas à primeira, constituam a *COLMÉIA*. A seguir, as tecnologias complementares que atendem aos requisitos restantes.

3.4.1 Tela de Retorno Tátil

Para atender o requisito de *retorno*, será incorporada ao dispositivo uma tela de retorno, que consiste basicamente em um plano colocado ao fundo do periférico que, na obstrução de um sinal, há necessariamente o toque do usuário na tela. Essa simulação de toque auxilia o usuário no uso do periférico, pois retorna uma sensação tátil relacionada à obstrução. Outras forma de retorno, sonora e visual, cabem ao *device driver* implementar a fim de melhorar a interação homem-máquina.

3.4.2 Correção de Erros de Acesso

Devido à implementação de vários transmissores/receptores de LEDs infravermelho, organizados em forma de matriz, podem ocorrer erros de acesso, segundo a especificação do próprio sistema. Para amenizar a taxa de erros decorrente desse conflito, e assim permitir o *signo composto* em níveis aceitáveis, serão utilizadas conjuntamente duas técnicas: taxa de varredura e ordenação de células, as quais serão descritas nos capítulos seguintes, referentes ao periférico e seu *device driver*, respectivamente.

4 *O Periférico*

Assim como fora apresentado no início deste trabalho, algumas funções da tecnologia serão implementadas em nível de *hardware* e outras em *software*. No primeiro caso, foram selecionadas algumas tecnologias para compor o periférico propriamente dito. A seguir, o projeto do periférico e suas tecnologias.

4.1 Os Componentes

Os componentes do dispositivo físico têm, em princípio, dois propósitos: aquisição de dados do usuário e sua transmissão ao computador. O primeiro utiliza por base o sistema por infravermelho, enquanto que o segundo a comunicação serial com o computador.

4.1.1 Dispositivo de Aquisição de Dados

O dispositivo de aquisição de dados consiste em uma matriz delimitada por emissores/receptores infravermelho, para a aquisição bruta dos dados, e uma tela de retorno tátil para fins de acessibilidade. Esse conjunto é montado sobre uma estrutura rígida para garantir as devidas dimensões do modelo. O primeiro componente, a matriz, funciona através da leitura, por varredura seqüencial, dos receptores de luz infravermelha, os quais se encontram dispostos em forma de L na margem da matriz¹ e lateralmente opostos ao seus transmissores correspondentes. No ato da leitura, os receptores que não foram incididos com luz infravermelha acusarão, na prática, um objeto que obstruiu o sinal vindo do seu respectivo transmissor. Dessa forma, é possível assinalar várias obstruções simultâneas, ao contrário do previsto nas telas sensíveis ao toque de mesma tecnologia - que após detectada uma obstrução a envia imediatamente ao seu controlador (*device driver*).

¹Não necessariamente na mesma orientação que a letra L.

Os receptores, numerados em seqüência, têm seus valores (obstruído = 1 / não obstruído = 0) transmitidos ao computador. Cabe ao computador refazer a matriz, com base na numeração dos receptores, e calcular os pontos de acesso (esse cálculo será visto no capítulo seguinte, referente ao *device driver*).

A Figura 6 ilustra um exemplo de receptores numerados e a direção dos raios infravermelhos. Os receptores estão assim numerados por entender, inicialmente, que o acesso parte de cima para baixo (ordem de leitura para linguagens latinas) e da direita para a esquerda (beneficiando o acesso proveniente da mão direita). Assim, caso haja múltiplos acessos, dá-se a preferência àquele provocado mais à direita. Essa ordem, obviamente, pode ser alterada pelo *device driver* segundo as preferências do usuário.

	↓	↓	↓	↓	↓	↓	↓	↓	↓	
1										←
2										←
3										←
4										←
5										←
6										←
7										←
8										←
	9	10	11	12	13	14	15	16		

Figura 6: Exemplo de matriz 8x8

A segunda parte do dispositivo, a tela de retorno tátil, tem por função oferecer uma base de apoio para o mapa de signos. Esse plano é posicionado de tal forma que, à ocorrência de uma obstrução de sinal, ocorre o toque na tela, seja esse toque direto com o dedo ou através de um objeto indicador, provendo uma sensação tátil como resposta ao acesso. Como a tela de retorno tátil não possui restrições de material para a sua confecção, pode tanto ser produzida utilizando uma placa rígida - uma peça de plástico ou metal - como outros materiais mais maleáveis, tais como gel ou silicone.

Dimensões

Como a idéia da *COLMÉIA* é permitir a troca do mapa de signos - o qual se encontra circunscrito pelos diodos infravermelho - têm-se como dimensões do mesmo 210 mm de altura por 297 mm de largura, ou uma folha A4 em modo paisagem, a fim de facilitar o seu uso utilizando folhas comuns de desenho e garantindo, dentre outros fatores, baixo

custo e facilidade na criação e alteração dinâmica de mapas. Entretanto, como a folha será fixada à estrutura do periférico - que assegurará a sua posição para o uso - serão subtraídos de cada lado 10 mm para essa fixação, restando 190 mm de altura por 277 mm de largura para a construção do mapa de signos.²

Quanto à granularidade da matriz³, devem-se levar em conta dois fatores:

1. A resolução é diretamente proporcional ao seu custo, devido ao número de diodos instalados ao longo do perímetro da matriz;
2. O acesso do usuário deve acionar ao menos uma célula para garantir retorno lógico (visual ou sonoro) no acesso.

No teclado convencional, as teclas devem possuir dimensões e distâncias mínimas para evitar o toque indesejado de teclas vizinhas, exatamente o inverso do caso acima especificado. Utiliza, pois, um mínimo de 19,1 mm de distância entre os seus centros para evitar esse tipo de problema. Usando esse valor como referência para a *COLMÉIA*, será utilizada aproximadamente a metade desse valor para definir as dimensões das células: 9,2 mm. Com isso, cada acesso terá pelo menos uma linha e uma coluna obstruída e, portanto, uma célula acessada - garantindo o retorno lógico. Além disso, também garante que haverá, no máximo, duas linhas e duas colunas acessadas, ou quatro células. Esse último, o qual caracteriza um múltiplo acesso, será discutido e resolvido mais adiante na seção 4.2.

O valor exato de 9,2 mm foi definido, além dos fatores já mencionados, para conceber 30 células por linha e 20 por coluna, totalizando 600 células. Contudo, esse número é experimental, somente com um estudo de caso posterior é que será possível avaliar o seu impacto na acessibilidade do usuário. Por enquanto, o valor está justificado para atender a um aspecto técnico e de facilidade de implementação.

Cada diodo será posicionado no meio da lateral da célula. Dessa forma, cada uma das células da matriz será dividida em quatro partições iguais, seccionadas pelos raios vertical e horizontal: o seu centro está calibrado para ser o centro da área de acesso. Em relação ao terceiro eixo, altura, os diodos serão posicionados a 3 mm acima da tela de retorno tátil, a fim de garantir o toque na tela à obstrução do raio infravermelho. Assim como o

²Outro motivo é que as impressoras, em geral, não conseguem imprimir com uma margem inferior a 10 mm.

³Quantidade de células definidas em uma área delimitada

tamanho da célula, esse número também é experimental, vista a originalidade do projeto em termos de uso e tecnologias agregadas. Por esse motivo, a folha contendo o mapa de signos deverá ser fixada em uma estrutura de encaixe bastante rígida e sem folgas. A Figura 7 apresenta o dispositivo e suas camadas: matriz de diodos (1), mapa de figuras (2) e tela de retorno tátil (3).

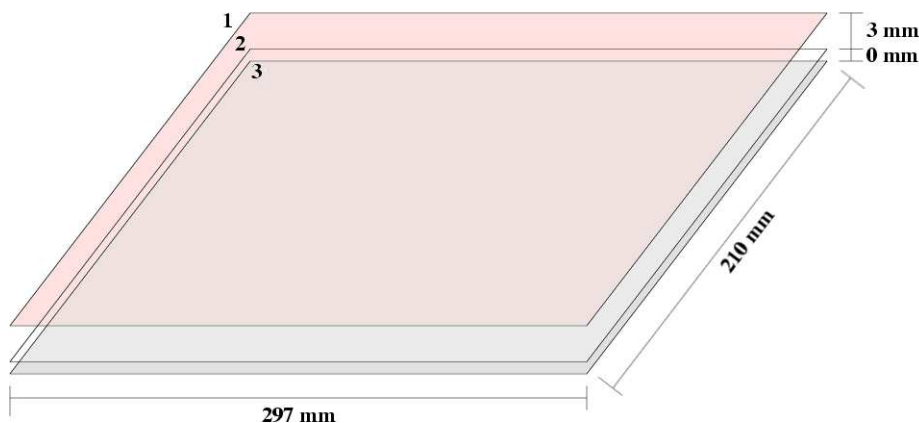


Figura 7: O dispositivo e suas camadas

4.1.2 Comunicação com o Computador

Os dados adquiridos pelos receptores infravermelho serão, posteriormente, transmitidos em seqüência ao computador através de uma comunicação serial. Esse tipo de comunicação, além do seu baixíssimo custo de implementação, é encontrado em praticamente todos os computadores do tipo PC para transmissão de dados a baixas velocidades (na ordem de Kbps), não requerendo a compra de *hardware* adicional para o seu uso. Além disso, um cabo de comunicação serial entre 1,5 m e 3 m é suficiente para conectar os componentes no nível físico, permitindo o seu uso ao lado de outros periféricos de entrada de dados como teclado e *mouse*.

Assim como qualquer outro tipo de comunicação, é necessário estipular alguns parâmetros para controle da transmissão lógica dos dados. Em primeiro lugar, a sincronização da comunicação: devido ao fato dos dados serem enviados sempre em mesma quantidade (o valor de todos os receptores) e em um intervalo regular de tempo, assume-se uma transmissão regular dos dados. Isso permitirá uma transmissão compassada de dados sem a necessidade constante de sincronização da comunicação - o que poderia nesse caso até mesmo atrasá-la (por causa das mensagens de controle de transmissão e recepção).

O Protocolo de Comunicação

Além dos dados propriamente ditos, serão necessários alguns bits adicionais para garantir o “compasso” da comunicação. Esse conjunto de bits, denominado *preâmbulo*, marca o início dos dados a serem processados. Para o *device driver* reconhecê-lo, é atribuído um valor padrão de bits de difícil ocorrência em outro trecho de cada pacote de dados: 10101010. Essa seqüência, quando recebida pelo *device driver*, será facilmente identificada e preparará o programa a receber o conjunto de dados referentes aos fotorreceptores.⁴

Há também bits reservados, que poderão futuramente servir para outros propósitos (vislumbram-se outras formas de utilizar o dispositivo, como por exemplo utilizar diversas unidades de *COLMÉIA* conectadas a um único computador - devidamente identificadas através de um código ou ID), têm seu espaço reservado após os dados. Atualmente, por não possuir utilidade prática, será preenchido de 0s (zeros). O protocolo completo de comunicação pode ser visualizado na Figura 8.

<i>Preâmbulo</i> (8 bits)	<i>Dados</i> (50 bits)	<i>Bits reservados</i> (6 bits)
------------------------------	---------------------------	------------------------------------

Figura 8: Protocolo de comunicação serial

Assim, o *preâmbulo* contendo 8 bits, o campo de *dados* 50 e os *bits reservados* 6, chega-se ao valor de 64 bits para o pacote completo de dados que é transmitido em um intervalo regular de tempo. A partir desse número, é possível calcular a velocidade de transmissão entre o dispositivo e computador através da comunicação serial.

Esse protocolo, assim definido, traz uma grande facilidade técnica: dispensam-se os comandos ou processos de iniciação (ou inicialização) do dispositivo físico - como ocorre com o teclado, por exemplo - quando esse é conectado ao computador. Isso significa que, uma vez em operação o *device driver* da *COLMÉIA*, o periférico pode ser conectado e desconectado do computador sem implicar em problemas decorrentes dessa prática (efeito *plug'n'play*).

⁴Baseado no preâmbulo do protocolo de quadro Ethernet

Velocidade de Transmissão

Quanto à velocidade de transmissão dos dados ao computador, deve ser estabelecido um valor que atenda às necessidades do dispositivo sem comprometer o rendimento global do computador - por excesso de carga. Nas telas sensíveis ao toque, a taxa de envio dos pacotes ao computador é de 50 pacotes por segundo, ou um a cada 20 ms (3M, 2001). Utilizando esse valor como parâmetro para pacotes de 64 bits, é necessária uma velocidade mínima de 3200 bps. Como a comunicação serial em PCs baseia-se, geralmente, em múltiplos de 1200 (1200, 2400, 4800, 9600 e 19200), utilizar-se-á, a priori, o próximo valor da escala: 4800 bps. Assim, garante-se até mesmo uma margem contra falhas eventuais de comunicação, basta sincronizar uma primeira vez os bits para, em seguida, utilizar o meio de entrada de dados conforme a configuração preestabelecida.

4.2 Correção de Erros de Acesso

Assim como em qualquer outro sistema constituído de uma matriz de emissores/receptores e ativado por obstrução (o presente estudo contempla apenas um tipo por considerar os demais não enquadrados no escopo do projeto), este também apresenta sérias restrições de signo composto por causa da estrutura das células em matriz. A perda momentânea de acessos pode inviabilizar o uso corriqueiro desse tipo de periférico, conforme demonstra a equação 4.1.

$$perda \leq (acessos - 1) * 2 \quad (4.1)$$

Na prática, significa que a cada acesso adicional, podem ser “sombreados” outros dois pontos de acesso, identificando de forma equivocada dois acesso não realizados, totalizando quatro pontos.

Na Figura 9, os pontos 2.14 (i) e 6.11 (ii), quando acessados simultaneamente, podem gerar outros dois pontos de acesso: 2.11 e 6.14, formando exatamente os vértices de um retângulo.

Embora pareça bastante raro o caso de três ou mais acessos simultâneos, isso pode acontecer. Tomemos como exemplo uma caneta como objeto apontador: se essa for colocada em posição horizontal ao plano e diagonalmente sobre a matriz, ao invés de marcar

	↓	↓	↓	↓	↓	↓	↓	↓	↓	
1										←
2			iii			i				←
3										←
4										←
5										←
6			ii			iv				←
7										←
8										←
	9	10	11	12	13	14	15	16		

Figura 9: Dois acessos simultâneos no sistema por infravermelho.

	↓	↓	↓	↓	↓	↓	↓	↓	↓	
1										←
2										←
3		ix	viii	vii	vi	i				←
4		xiii	xii	xi	ii	x				←
5		xvii	xvi	iii	xv	xiv				←
6		xxi	iv	xx	xix	xviii				←
7		v	xxv	xxiv	xxiii	xxii				←
8										←
	9	10	11	12	13	14	15	16		

Figura 10: Múltiplo acesso provocado por objeto: o retângulo 3.10-7.14.

uma semi-reta em diagonal estará delimitando o acesso de todo um retângulo onde dois de seus vértices são as extremidades do objeto. A Figura 10 ilustra esse exemplo, onde se forma o quadrado de 25 acessos 3.10 a 7.14.

Uma das possíveis soluções para resolver esse problema, em nível de *hardware*, é aumentar a taxa de varredura para evitar o acesso simultâneo, coletando assim os acessos nas células em um tempo curto o bastante para evitar o acesso combinado simultâneo sem comprometer o acesso combinado seqüencial. Assim como no teclado convencional, estabelece-se uma ordem de varredura e, em seguida, essa varredura é feita ordenadamente, prevenindo que duas células, ou teclas, sejam acionadas num mesmo intervalo de tempo, preservando o acesso por tempo indeterminado. Para ilustrar essa situação em um teclado, tem-se a tecla Shift que, uma vez acionada, pode ficar indefinidamente pressionada. Em seguida, com a tecla a também pressionada, é enviado o símbolo A ao SO. Durante todo esse processo, Shift manteve seu estado original sem sofrer alterações oriundas das outras teclas, inclusive de a. Ao transportar essa mesma situação para o

sistema por infravermelho, não será possível, sem uma devida filtragem de erros, manter o estado de Shift inalterado.

Como o valor estimado inicialmente para a velocidade de transmissão foi de 50 pacotes por segundo a 4800 bps, para garantir o acesso combinado apenas seqüencial será estipulado o tempo de 100 pacotes por segundo a 9600 bps, duplicando a transmissão de dados e minimizando as possibilidades de acessos simultâneos. No capítulo seguinte, referente ao *device driver* da **COLMÉIA**, será abordada a segunda parte dessa solução.

5 *O Device Driver*

O *device driver* é uma camada de *software* que integra *hardware* e sistema operacional (SO). Sua função básica é atender às requisições do dispositivo físico e enviá-los ao SO segundo as suas regras e prioridades. No caso da *COLMÉIA*, servirá como um agente de controle, atuando na recepção e tratamento dos dados adquiridos pelo periférico - cujos processos serão detalhados ao longo deste capítulo.

5.1 **Recepção dos Dados**

O periférico, assim como fora descrito no capítulo anterior, envia regularmente uma quantidade fixa de bits ao computador, a qual foi denominada *pacote de dados*, sem qualquer tipo de tratamento da informação contida. Cabe ao *device driver*, portanto, tratar devidamente os dados, codificando e traduzindo-os para os valores válidos usados na interação homem-máquina corrente - conforme a configuração prévia do usuário. Esse processo de recepção e, numa segunda etapa, interpretação, segue alguns passos a serem executadas em seqüência para cada pacote, sendo abaixo descritas em sua ordem de processamento.

Sincronização da Transmissão

A primeira tarefa consiste em reconhecer o padrão dos bits que estão sendo enviados pelo periférico, pois uma vez energizado o periférico, esse passa a transmitir regularmente ao computador. Tem-se que o protocolo de comunicação dos dados possui três campos: *Preâmbulo*, *Dados* e *Bits Reservados*, onde o primeiro é um padrão de bits 1 e 0 alternados exatamente para delimitar o começo de um pacote de dados. Dessa forma, após o *device driver* receber e detectar o padrão 10101010 (o *Preâmbulo*), sabe-se que os próximos 50 bits pertencem ao campo *Dados* e, portanto, sincroniza-se o próximo bit com o primeiro

bit do campo *Dados*.

Seleção dos Bits Válidos

Uma vez detectado o início do campo *Dados*, armazenam-se os próximos 50 bits recebidos em uma área de memória temporária (ou *buffer*) referente a esse campo. Futuramente, os 6 bits seguintes de *Bits Reservados* também serão armazenados, contudo por enquanto não têm utilidade prática - sendo então descartados. A informação útil, antes volátil, agora está armazenada em memória para posterior manipulação do seu conteúdo.

Conversão dos Bits Seqüenciais em Matriz

O campo *Dados* armazena os valores (em binário) dos 50 fotorreceptores dispostos ao longo da matriz. Contudo, essa informação é enviada pelo periférico em formato linear, seqüencial. Para que o *device driver* possa detectar os acessos de células (linha x coluna), é necessário pré-processar essa informação, convertendo-a do formato linear para o formato de uma matriz - no caso da *COLMÉIA*, uma matriz de 30 colunas por 20 linhas. Esse pré-processamento servirá para reconstruir a matriz original do periférico.

Exemplo 1: Matriz 8x8 com Múltiplo Acesso

Para ilustrar a conversão, tem-se o seguinte exemplo de matriz no dispositivo de aquisição de dados na Figura 11.

	↓	↓	↓	↓	↓	↓	↓	↓	↓	
1										←
2										←
3										←
4			ii		i					←
5										←
6										←
7			iv		iii					←
8										←
	9	10	11	12	13	14	15	16		

Figura 11: Exemplo de matriz 8x8 com múltiplo acesso

Nesse exemplo, há um múltiplo acesso nas células 4.11, 4.13, 7.11 e 7.13. Ao trans-

mitir esses dados para o computador, através de uma comunicação serial, linearizou-se essa informação: o periférico envia o estado de obstrução (1 ou 0) de cada fotorreceptor, conforme a Figura 12.

0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)

Figura 12: Exemplo de transmissão serial de 16 bits (matriz 8x8)

Para reconstruir a matriz, recorre-se às especificações do dispositivo para identificar o número de colunas e de linhas. Para a **COLMÉIA**, são 30 colunas e 20 linhas; para o exemplo apresentado nesse capítulo, 8 colunas e 8 linhas. Uma vez tendo esses números, monta-se a matriz com tais dimensões, onde o bit 1 representa a coluna/linha obstruída e 0 não obstruída. Aplica-se, em cada célula, uma função lógica **E** aos valores coluna e linha. Quando houver coluna e linha obstruída (1), a célula terá valor 1, caso contrário 0. A Figura 13 apresenta essa conversão.

1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16

Figura 13: Exemplo de matriz 8x8 reconstruída

Dessa forma, a matriz de colunas/linhas obstruídas é reconstituída em *software* para permitir posteriores tratamentos de dados, implementados através de módulos adicionais - imprescindível para isolar as células, apesar de consumir mais memória que o formato anterior (cerca de doze vezes mais).

5.2 Interpretação dos Dados

O *device driver*, aqui proposto, está organizado em duas camadas: recepção e interpretação dos dados (Figura 14). A primeira camada serve de base para a segunda, cujo propósito é

filtrar essa informação recebida e, assim, enviar ao SO somente os signos desejados pelo usuário sem qualquer desvio na entrada de dados ao computador.

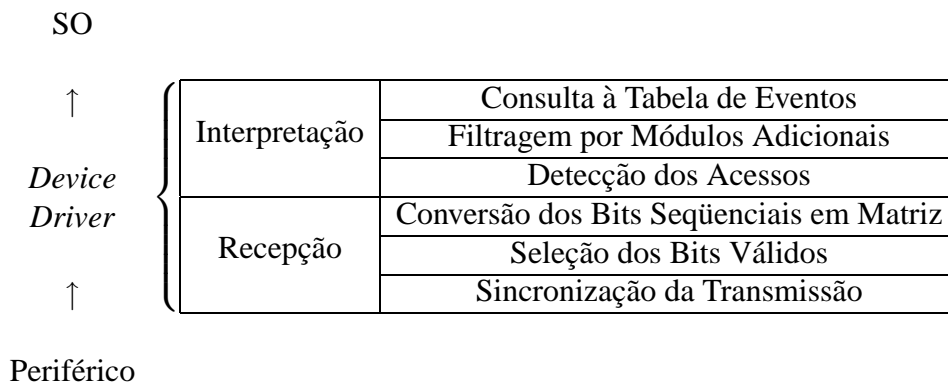


Figura 14: Divisão do *device driver* em camadas

Detecção dos Acessos

Em um teclado convencional, há basicamente dois tipos de eventos possíveis para cada tecla: pressionamento (*key press*) e liberação (*key release*). O primeiro ocorre ao pressionamento de uma tecla em repouso; o segundo, a liberação da tecla pressionada. Entre esses dois eventos, há uma relação temporal direta: o segundo evento só é possível após a ocorrência do primeiro. Na **COLMÉIA**, as relações temporais também são contempladas, ampliando as funcionalidades de uso (além de manter a compatibilidade com o teclado e o *mouse*).

Para permitir as relações temporais ao nível do *device driver*, é necessário um histórico de ações prévias que, uma vez catalogadas, poderão trazer os subsídios necessários para tal procedimento. Contudo, como o periférico envia ao computador toda a matriz de células ao invés de um único acesso (pressionamento ou liberação), será necessário arquivar todo esse conjunto de dados para ser possível essa funcionalidade do periférico. Portanto, devem ficar registradas pelo menos duas matrizes além da matriz corrente (a última enviada pelo periférico), aqui denominada *M1*:

1. A última matriz diferente da atual (*M2*): armazena-se, além da matriz corrente, a última matriz onde houve pelo menos uma célula com valor alterado. Na prática, significa que é armazenada a última ação do usuário (pressionamento ou liberação);

2. A penúltima matriz diferente da atual ($M3$): além da última matriz, também armazena-se a penúltima matriz diferente. Essa, em especial, será utilizada para identificar eventos sensíveis ao contexto, como por exemplo o duplo clique.

Associada a cada uma dessas duas matrizes, será também armazenado o exato momento em milissegundos em que foi recebida a matriz pelo computador (*timestamp*). Assim como a matriz informará **qual** a célula modificada, esse valor informará **quando** houve uma ação provocada pelo usuário e percebida pelo periférico. Dessa forma, será possível manter a compatibilidade com o teclado ao processar a repetição de signos (*rate*) e o tempo de espera (*delay*), além de fornecer dados suficientes para outros padrões de manipulação de dados por parte dos módulos adicionais.

Com isso, tem-se assim a terceira dimensão para a manipulação dos dados: o tempo. A título de exemplo, esse último recurso pode ser utilizado para casos mais específicos de uso: uma vez escolhida a figura, mantém-se a célula selecionada por um tempo mínimo, em média 1 (um) segundo, para o *device driver* distinguir entre a leitura e a escolha efetiva de um signo - possibilitando “navegar” por sobre o mapa sem implicar acessos indesejados¹.

Após colher as matrizes anteriores e a matriz corrente, identificam-se as células cujo valor foi modificado para, em seguida, repassá-las para o próximo estágio do *device driver*. Esse repasse será feito através de um vetor de células, o qual informa:

- A quantidade total de células modificadas;
- A variação de cada célula:
 1. 0 = célula liberada: teve seu valor alterado de 1 em $M2$ para 0 em $M1$;
 2. 1 = célula acessada pela primeira vez: teve os valores 0, 0 e 1 nas matrizes $M3$, $M2$ e $M1$, respectivamente;
 3. 2 = célula acessada pela segunda vez consecutiva: foram identificados os seguintes valores em cada matriz: $M3 = 1$; $M2 = 0$; $M1 = 1$;
- Posição vertical de cada célula;
- Posição horizontal de cada célula.

¹Essa implementação de acesso temporal já existe em teclados virtuais como o GOK (THE GNOME PROJECT, 2003b), parte do projeto GNOME (THE GNOME PROJECT, 2003a), e o teclado virtual do Windows XP®.

Exemplo 1: Detecção dos Vários Acessos Simultâneos

O exemplo apresentado no item anterior, o qual mencionava quatro acessos, será utilizado ao longo do capítulo para demonstrar todo o processo de tratamento da informação recebida pelo periférico. Como nessa etapa de detecção de acessos, além do acesso regular a uma célula, também é possível discriminar “duplos cliques” ou liberações de célula, será considerada - para fins didáticos - uma liberação da célula 2.11 como um quinto acesso detectado pelo *device driver*. A seguir, a Figura 15 ilustra o vetor contendo as cinco células modificadas: 2.11, 4.13, 4.11, 7.13 e 7.11.

5	0 2 11	1 4 13	1 4 11	1 7 13	1 7 11
---	------------	------------	------------	------------	------------

Figura 15: Vetor de células modificadas

O vetor, portanto, apresenta a seguinte estrutura: o primeiro valor, 5, informa a quantidade de células modificadas; os valores seguintes, organizados em trios, representam o valor e as posições vertical e horizontal de cada célula.

Após esse envio das células e seus valores para filtragem dos módulos adicionais, executa-se um rodízio das matrizes, visto que a nova matriz $M1$ passa a ser a última matriz modificada. Assim, descarta-se a matriz $M3$ ao se atribuir $M2$ a $M3$ e, numa etapa seguinte, $M1$ a $M2$, conforme a Figura 16.

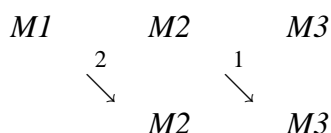


Figura 16: Rodízio das matrizes armazenadas em memória volátil

5.2.1 Filtragem por Módulos Adicionais

Resumidamente, o papel do *device driver* é receber, detectar, manipular e enviar os signos para o SO. Até essa etapa, foram realizados os dois primeiros processos. Para o processo seguinte, de manipulação, fica a cargo dos módulos adicionais tratarem a informação a fim de adequá-la segundo as configurações do usuário.

Os módulos serão organizados em uma fila, cuja ordem de prioridade se dá pela posição na mesma. Por isso, é preciso, por parte do usuário, cuidado tanto na seleção

quanto na ordenação dos módulos. A única restrição em relação à escolha dos módulos está no primeiro elemento da fila: este é obrigatoriamente o de seleção e ordenação de acessos (*SOA*), pois devido ao sistema por infravermelho ser passível de erros por “sombreamento” de células², é imprescindível corrigir esse desvio antes de fornecer os dados aos outros módulos, como demonstra a Figura 17.

Dados brutos →

<i>SOA</i>	Módulo 2	Módulo 3	...
------------	----------	----------	-----

 → Dados filtrados

Figura 17: Filtragem dos dados pelos módulos adicionais

Além do módulo básico *SOA*, o *device driver* oferecerá toda uma interface de funções (*Application Program Interface* - *API*) aos outros módulos. Esse recurso vai permitir que a tecnologia seja altamente expansível e adaptável a diversas outras funcionalidades, que podem ser livremente agregadas à *COLMÉIA* sem qualquer reimplementação do *device driver*, bastando que os novos módulos atendam a essa *API* para a plena comunicação e funcionamento de todos os componentes em *software*. Cabe aqui ressaltar que nenhum módulo acessará diretamente o *SO*: uma vez tratados, os signos serão retornados para o *device driver* e, posteriormente, enviados ao *SO*.

Primeiro Módulo: Seleção e Ordenação de Acessos

Aliado à alta taxa de varredura, ao nível do *device driver* será implementado um método de detecção de conflitos de acessos, estabelecendo seleção e prioridades ou ordem dos acessos, trazendo assim uma solução para esse problema. Por exemplo, num múltiplo acesso com duas células, pode-se estabelecer apenas o primeiro acesso como válido ou aceitá-los sequencialmente em uma ordem preestabelecida. Essa funcionalidade será implementada como um módulo do *device driver*, dada a sua natureza de manipulação dos dados.

O conceito básico do módulo é bastante simples. Primeiramente, o usuário configura duas variáveis:

1. *Ordem*: define a seqüência em que devem ser ordenados os signos que foram acessados simultaneamente. Possíveis valores: ↙, ↘, ↗, ↖. No caso padrão (↙), serão ordenados os acessos de cima para baixo e da direita para a esquerda;

²Argumentação vista na seção 4.2

2. *Seleção*: permite os valores todos (padrão), primeiro ou último. Significa que será repassado ao SO somente o(s) acesso(s) aqui especificado(s) - em conformidade com a ordem preestabelecida.

Exemplo 1: Aplicação de uma Política SOA

Feitas as devidas configurações, o módulo fará a filtragem dos acessos segundo a política especificada pelo usuário. Para melhor compreensão, utilizar-se-á o exemplo 1, já ilustrado na Figura 13 (página 51), onde nota-se claramente a necessidade de uma política de ordenação e seleção de acessos. Para esse caso em especial, foi definida a política de ordenação \surd e todos (valores padrão).

1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16

Figura 18: Exemplo de seleção e ordenação de múltiplo acesso

Nesse exemplo, onde houve quatro acessos simultâneos (a liberação da célula 2.11 é ignorada por não ser um acesso regular do tipo pressionamento, tendo portanto um tratamento diferenciado), não é possível afirmar se houve ou não “sombreamento” das células, pois as quatro células formam exatamente um retângulo. Assim, fica a cargo do *device driver* filtrar e tratar essa informação com base na configuração do módulo SOA, conforme descreve o algoritmo a seguir.

1	<i>Identificar a primeira liberação de célula L por varredura seqüencial</i>
2	<i>Estabelecer como matriz de busca a submatriz M que ainda não foi processada no passo 1</i>
3	<i>Retornar ao passo 1, tendo como matriz de busca M, até M não conter células a serem analisadas</i>
4	<i>Assumir como matriz de busca M a matriz original (completa)</i>
5	<i>Identificar o primeiro acesso AA por varredura seqüencial</i>
6	<i>Remover todos os acessos dispostos em mesma coluna ou linha de AA</i>
7	<i>Estabelecer como matriz de busca a submatriz M que ainda não foi processada nos passos 5 e 6</i>
8	<i>Retornar ao passo 5, tendo como matriz de busca M, até M não conter células a serem analisadas</i>
9	<i>Assumir como matriz de busca M a matriz original</i>
10	<i>Identificar o primeiro acesso A por varredura seqüencial</i>
11	<i>Remover todos os acessos dispostos em mesma coluna ou linha de A</i>
12	<i>Estabelecer como matriz de busca a submatriz M que ainda não foi processada nos passos 10 e 11</i>
13	<i>Retornar ao passo 10, tendo como matriz de busca M, até M não conter células a serem analisadas</i>

Transpondo para o exemplo dessa seção, tem-se que a primeira e única liberação de célula identificada é 2.11 (Figura 20), sendo essa prontamente adicionada ao vetor de retorno. O primeiro acesso, identificado no passo 10, é 4.13 (Figura 21), pois é o que se encontra mais a nordeste na matriz³. Os acessos 4.11 e 7.13, dispostos em mesma coluna ou linha, são entendidos como o “sombreamento” indevido e são removidos (Figura 22). Para encontrar o segundo acesso, estabelece-se como matriz de busca a submatriz exatamente abaixo e à esquerda de 4.13, conforme a política de ordenação e seleção de acessos (Figura 23). Em seguida, identifica-se 7.11 como o segundo acesso (Figura 24). Uma nova submatriz é gerada (Figura 25), e o processo finaliza por não haver mais qualquer tipo de acesso na matriz.

O módulo, que havia recebido o vetor de cinco acessos, retornará portanto ao *device driver* apenas 3 acessos - já ordenados:

³A política de ordenação \swarrow inicia na primeira linha e última coluna (nordeste) até chegar à última linha e primeira coluna (sudoeste).

3	0	2	11	1	4	13	1	7	11
---	---	---	----	---	---	----	---	---	----

Figura 19: Vetor de células modificadas filtrado pelo módulo *SOA*

1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16

Figura 20: Identificação da primeira e única liberação de célula: 2.11

1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16

Figura 21: Identificação do primeiro acesso: 4.13

1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16

Figura 22: Remoção das células “sombreadas” 4.11 e 7.13

5	0	0	0	0
6	0	0	0	0
7	0	0	1	0
8	0	0	0	0
	9	10	11	12

Figura 23: Estabelecimento da submatriz M de busca

5	0	0	0	0
6	0	0	0	0
7	0	0	1	0
8	0	0	0	0
	9	10	11	12

Figura 24: Identificação do segundo acesso: 7.11

8	0	0
	9	10

Figura 25: Estabelecimento da submatriz M de busca

Outros Módulos

Uma das inovações que a *COLMÉIA* traz aos teclados de conceitos é a possibilidade de agregar, dinamicamente, novas e diversas funcionalidades e “comportamentos”. Isso significa, em termos práticos, que a tecnologia pode ser ampliada e adaptada conforme as necessidades (especiais ou não) do usuário. Essa capacidade de se “moldar” ao usuário no que tange aos seus recursos é viabilizada pelo *device driver* através de uma API, a qual possui um conjunto de funções para permitir a comunicação bidirecional de dados (envio de dados brutos e recebimento de dados filtrados).

As possibilidades são muitas. Sob a óptica da usabilidade, é possível implementar, em adição ao retorno tátil do periférico, um retorno visual e/ou sonoro, podendo assim emular um teclado convencional⁴ - com a vantagem dos botões e sons serem passíveis de modificação em tempo real. Cada acesso implica, portanto, no retorno de duas células: a primeira é o próprio acesso; a segunda, uma célula reservada que está relacionada a um som⁵.

Em relação ao tempo, pode-se utilizar esse tipo de informação como valor de referência no acesso às células. Variáveis como tempo de espera e repetição de signos podem ser livremente alterados, assim como permite a criação de novos valores, tais como:

- *Tempo mínimo de acesso*: é necessário o acesso a uma célula por um tempo mínimo para caracterizar o acesso tal qual o pressionamento de um botão;
- *Reforçador*: a cada intervalo de tempo n que é mantido o acesso a uma célula, essa mesma célula assume diferentes “comportamentos”. Por exemplo: acessar a célula 26.4 durante 1 segundo representa “a”, entre 1 e 2 segundos “A”, e mais de 2 segundos o valor muda para “@”.

5.3 Consulta à Tabela de Eventos

Após os dados terem sido devidamente tratados pelos módulos adicionais do *device driver*, a última tarefa a ser realizada é enviar ao SO o signo correspondente à(s) célula(s)

⁴O aplicativo de troca de mensagens ICQ (MIRABILIS, 2004), por exemplo, implementa esse recurso de retorno sonoro simulando uma máquina de escrever como resposta à digitação.

⁵A perda de uma célula para esse fim é justificável pela significativa quantidade total de células da matriz: 600.

acessada(s).

A *COLMÉIA*, por suas propriedades, pode agregar várias células em torno de um mesmo signo. Por isso, foi criado um eficiente de armazenamento dessas informações em memória: para relacionar os elementos, será utilizada uma matriz (*MI*) onde cada célula contém um número, o qual corresponde ao índice de uma lista auxiliar de signos (*LAS*) - ambas definidos pelo usuário utilizando uma ferramenta gráfica de auxílio: a aplicação base. Assim, é possível associar várias células (inclusive sem qualquer relação espacial) a um mesmo signo sem perda excessiva de memória e processamento - e conseqüente menor desempenho. As Figuras 26 e 27 ilustram essa situação com uma matriz 8x8 para fins didáticos.

1	0	0	0	0	0	0	0	5
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	3	3	3	3	0	0	0
5	0	3	3	0	0	1	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	4	2	2	0
8	0	0	0	0	0	2	2	0
	9	10	11	12	13	14	15	16

Figura 26: Matriz de índices *MI*

0	
1	a
2	↑
3	casa
4	som 440 1
5	ação desligar

Figura 27: Lista auxiliar de signos *LAS*

O signo de número 0, cujo valor não pode ser alterado, representa nenhum signo; ou seja, não há qualquer signo relacionado a célula (vazia). Os outros signos, que podem ser até 600 distintos (um para cada célula), podem ser:

- *Simples*: uma série de n caracteres do teclado ou de eventos do *mouse* concatenados;
- *Som*: um som que será emitido através do auto-falante (*speaker*) do computador. Tem variação de frequência (Hertz) e tempo (segundos);

- *Ação*: uma ação a ser executada através do sistema operacional (copiar, colar, desligar o computador);

Cada um dos tipos, acima descritos, possui uma sintaxe específica - a fim de ser corretamente reconhecido pelo *device driver*. Esses três padrões, todos expressões regulares⁶, foram projetados para facilitar o seu uso por parte do usuário: fácil assimilação e estruturação lógica.

O tipo mais básico, *simples*, é formado por uma cadeia de caracteres, cujo conjunto está definido na norma ISO 8859-15 (extensão da norma ISO 8859-1, de línguas latinas, que incorpora novidades como o Euro), e eventos do mouse:

- As oito direções referentes ao plano formado pelos eixos x e y ;
- As duas direções do eixo z ;
- Os três botões.

Os caracteres, que são o equivalente às teclas normais do teclado, podem ser combinados com teclas compositivas: Shift, Ctrl, Alt e Altgr. Para tanto, utilizar-se-á o sinal “+” para indicar a combinação de um caracter com uma compositiva. Exemplos: a, Ctrl+V, casa, ↑. Os sons, segundo tipo, são descritos da seguinte maneira: a palavra som, um ou mais divisores (um ou mais espaços em branco ou tabulações), a frequência (a relação com as notas musicais está listada na tabela 3 do Anexo I), um ou mais divisores e o tempo de execução (número real com até uma casa decimal compreendido entre os valores 1 e 60). Ex.: som 440 1. E, por último, o tipo ação: a palavra ação, um ou mais divisores e a sentença que define a ação imperativa a ser executada pelo SO. A lista com alguns exemplos de ações executáveis, bem como a sintaxe de uso, está descrita na tabela 1. A lista completa das ações reconhecidas pelo *device driver* se encontra no Anexo II.

Sintaxe da Ação	Descrição	Exemplo
desliga	Desliga o computador	ação desliga
copia	Copia a área de seleção	ação copia
execute <comando>	Executa uma aplicação	ação executa xcalc

Tabela 1: Exemplos de ações remetidas pelo *device driver* ao SO

⁶Um expressão regular define, formalmente, um conjunto de sentenças (*strings*) possíveis.

Exemplo 1: 5 Células, 1 Evento

Para finalizar o exemplo deste capítulo, aplica-se o vetor de acessos - já filtrado (Figura 19, página 58) à matriz de índices *MI* da Figura 26 (página 61). Desta forma, as células serão enfim traduzidas em signos e, em seguida, enviados ao SO - a última tarefa do *device driver*. A mescla do vetor e da matriz está descrita na Figura 28.

1	0	0	0	0	0	0	0	5
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	3	3	3	3	0	0	0
5	0	3	3	0	0	1	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	4	2	2	0
8	0	0	0	0	0	2	2	0
	9	10	11	12	13	14	15	16

Figura 28: Vetor aplicado à matriz *MI*

A primeira célula modificada, 2.11, não será consultada para enviar signos ao SO, visto que se trata de uma liberação de célula. Esta tarefa seria processada por um módulo adicional para tratar outros tipos de evento, como por exemplo repetição de signos. Em segundo, a célula 4.13; esta, em especial, refere-se ao signo casa (terceiro índice da lista auxiliar *LAS* - Figura 27). Por tratarem-se de caracteres de teclado, enviam-se esses quatro caracteres em seqüência, efeito semelhante à digitação em um teclado convencional. E, por último, a célula 7.11, que, assim como 2.11, não tem signo associado, sendo portanto ignorado. Ao final, é enviado ao SO apenas a seqüência de caracteres casa.

5.4 COLMÉIA: Independência da Aplicação

Enquanto que para o usuário o acesso a uma célula representa um signo enviado ao computador, para esse último - mais especificamente o SO - tal processo se assemelha a um evento originado pelo usuário: uma ação identificada pelo periférico (agente externo) que implica em uma reação do sistema (meio interno).

Esse esquema de atendimento a requisições para controle de interfaces - inspirado no modelo MVC (ATD, 2004) - é encontrado em diversas implementações, desde *software* de autoria como o Everest (COMPLEX, 2003) até ambientes gráficos (*desktop environments*),

como por exemplo o GNOME (THE GNOME PROJECT, 2003a)⁷. Essa é mais uma vantagem somada à *COLMÉIA*, pois livra as outras aplicações de qualquer adaptação adicional para serem compatíveis com a tecnologia. Os editores de texto, planilhas gráficas e navegadores *Web* já existentes no mercado, por exemplo, podem ser amplamente utilizados com a *COLMÉIA*, conjuntamente com o teclado e o *mouse*, sem aplicações intermediárias ou correções de código (*patches*).

⁷O GNOME, em particular, utiliza o próprio MVC de atendimento às requisições (SANCHES, 2003), herança da linguagem de programação *Smalltalk*.

6 *A Aplicação Base*

A *COLMÉIA* foi projetada para oferecer uma gama ampla de funcionalidades e configurações para diversos fins. Entretanto, é mister a incorporação, ao final, de uma ferramenta que apóie o usuário a definir correta e satisfatoriamente algumas variáveis que impactam diretamente o funcionamento do periférico e do *device driver*, mais especificamente em dois aspectos:

- Aspectos sensoriais: toda informação criada/modificada pelo usuário em termos visuais, táteis e sonoros;
- Aspectos “comportamentais”: todo processo de filtragem dos dados adquiridos pelo periférico e que pode modificar o seu conteúdo para atender a necessidades específicas do usuário em termos de acessibilidade.

Essa ferramenta se constitui de uma aplicação gráfica básica, desenvolvida com base em conceitos de usabilidade e acessibilidade. Sob esta óptica, pode-se concluir que as camadas de baixo nível da *COLMÉIA* (periférico e *device driver*) são responsáveis por *prover as possibilidades* de interação de homem-máquina, a partir dos recursos oferecidos pelo computador. As camadas de alto nível (encapsuladas na aplicação base) por sua vez, atuam como um *agente norteador* na escolha das opções consideradas mais adequadas às necessidades de cada usuário.

O resultado final desse processo, a configuração do periférico e do *device driver*, será denominado *perfil* do usuário, uma vez que essas configurações ajustam as camadas inferiores às necessidades específicas de um grupo de usuários.

Os conceitos de usabilidade e acessibilidade que orientam o desenvolvimento desta ferramenta foram definidos com base em (CYBIS, 2003). Esses conceitos se constituíram em premissas para o projeto de interface da aplicação base, e são listados a seguir:

1. *Condução*: a aplicação deve apresentar ao usuário uma linha de ação simples para atingir o seu objetivo. Nesse caso, configurar o periférico e o *device driver*;
2. *Brevidade*: aliado à premissa de *condução*, assegura que as ações do usuário devem ser curtas o bastante para serem executadas sem perda de informação durante o processo.
3. *Controle do usuário*: apesar da configuração se constituir em uma seqüência organizada de passos, deve-se dar ao usuário uma margem para escolher a ordem das subtarefas, bem como permitir-lhe navegar pelas tarefas já realizadas;
4. *Agrupamento*: a configuração se divide em dois grupos distintos e bem delimitados: *aspectos sensoriais* e *aspectos “comportamentais”*, conforme visto no início deste capítulo;
5. *Retorno*: para toda ação do usuário, deve haver uma resposta do computador, mantendo-o assim informado do estado da aplicação (executando uma ação, em espera, etc.);

Além disso, foram consultados também (ABNT, 2000) e (BENSON et al., 2002) para a especificação da interface. Em termos de acessibilidade, foi utilizado como referência as recomendações de (CAMPOS; SILVEIRA, 1998, p.15-20).

6.1 Interface da Aplicação

Esse é o elemento mais importante da aplicação base, visto que o seu propósito é de auxiliar o usuário na configuração do periférico e dos módulos do *device driver*, e que esse processo pode tornar-se complexo frente às necessidades do usuário. Por esse motivo, serão bastante discutidas ao longo deste capítulo as telas da aplicação e as transições entre as mesmas.

Na interface principal encontram-se as duas funções básicas da aplicação: prover meios para criar/alterar *perfis*, bem como recuperá-los para a sua utilização como objeto de interação homem-máquina. A aplicação é iniciada com a apresentação do menu inicial, conforma a figura 29.

Ao escolher a primeira opção, Criar um perfil, alterna-se esse menu pela janela de criação, a qual será responsável por apresentar todas as opções de configuração de um

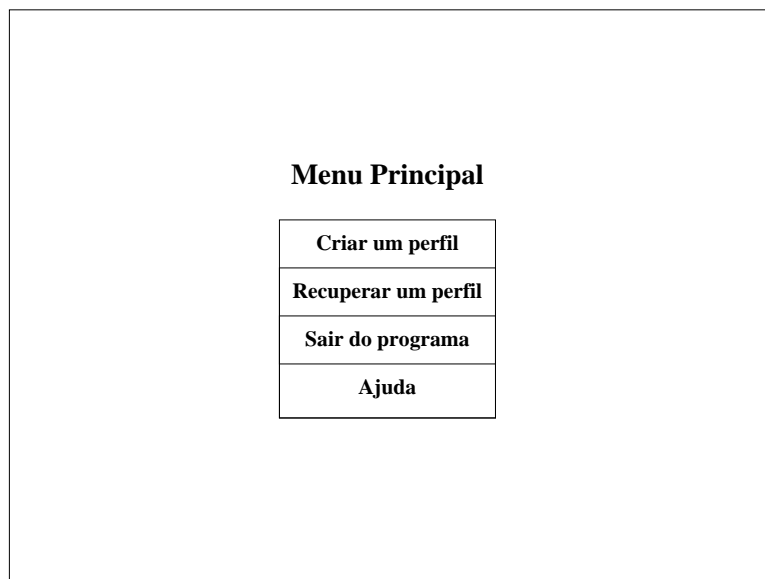


Figura 29: Menu principal da aplicação base

novo *perfil*. Caso a escolha seja a segunda, **Recuperar um perfil**, transporta-se o usuário para a janela de depósito de *perfis*, onde é apresentada uma lista dos *perfis* já construídos e armazenados no sistema. A última opção, de ajuda, explica alguns conceitos fundamentais como *perfil*, *depósito*, entre outros, além apresentar de dicas para utilizar a aplicação.

6.1.1 Janela de Criação

A interface de criação deve ser bastante acessível, especialmente porque nela será definido o **futuro** *perfil* construído pelo usuário; ou seja, durante esse processo de criação, deve haver meios simples e fáceis de usar para não sobrecarregar o usuário já nessa fase inicial de uso da ferramenta. O modo de interação escolhido foi o padrão assistente, sendo a ordem das tarefas proposta pela aplicação. Convém lembrar que essa ordem é recomendada, porém não obrigatória para permitir um maior controle do usuário sobre a situação. Cabe à aplicação, nesses casos, prevenir o usuário de possíveis problemas que poderão ocorrer caso decisões sejam tomadas precipitadamente. Nessa janela, foram enfatizadas as premissas de *condução* e *agrupamento* - a fim de orientar, de forma simples e objetiva, o usuário a concluir a tarefa.

A figura 30 ilustra a janela de criação da aplicação: uma moldura inferior, que apresenta a ordem das tarefas, bem como o estado de cada uma delas (realizada, não realizada,

etc.), e a moldura superior, a qual é dinamicamente alterada para apresentar a tarefa escolhida na moldura inferior.

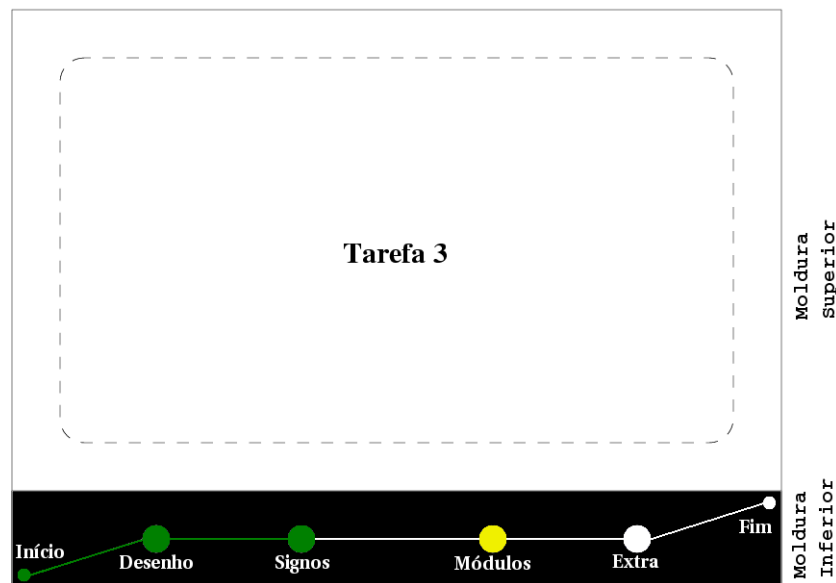


Figura 30: Janela de criação

A Moldura Inferior

A configuração completa da *COLMÉIA* é uma *ação atômica*¹: definir os parâmetros que serão usados na interação homem-máquina. Contudo, há nessa ação uma série de subtarefas, onde cada uma possui as suas características e dependências obrigatórias particulares, e devido a isso serão agrupadas em quatro blocos homogêneos e consecutivos:

1. ***Desenho do mapa de signos***: a primeira tarefa a ser realizada é o desenho do mapa de signos, utilizando para isso figuras representativas ou mesmo definindo conjuntos de células. Na prática, o usuário estará definindo os agrupamentos de células (da matriz) que servirão de pontos de acesso *real* dos signos usados na interação homem-máquina;
2. ***Atribuição dos signos***: em seguida, atribui-se a cada figura representativa (ou agrupamento de células) um signo, o qual pode ser *simples* (sequência de caracteres ou movimento do *mouse*), *som* (sons emitidos pelo computador) ou uma *ação* (ações

¹Uma vez iniciada, a ação só será considerada concluída quando *todas* as suas subetapas forem realizadas com sucesso.

realizadas pelo Sistema Operacional ou aplicação em execução). Tanto a definição das figuras quanto a sua relação com signos são subtarefas bastante simples, atendendo também à premissa de *brevidade*;

3. *Seleção e ordenação dos **módulos***: tratando agora do *device driver*, selecionam-se os módulos que farão o devido tratamento dos dados. O módulo obrigatório, *SOA*, não será apresentado - visto que não pode ser movido ou mesmo removido da lista;
4. *Configuração **extra** dos módulos*: alguns módulos podem implicar em configurações adicionais dos signos, como por exemplo a velocidade de repetição dos signos (equivalente ao do teclado convencional). Para tanto, será disponibilizada pela aplicação uma API para incorporação de interfaces de configuração dessas novas variáveis.

Como pôde-se notar, o grupo 2 depende que as subtarefas do grupo 1 tenham sido concluídas previamente. Por isso, estabelece-se uma relação unilateral de dependência entre ambos. Espera-se, com isso, ter subsídios para informar o usuário do bom andamento das tarefas (premissa de *retorno*). O mesmo ocorre com o grupo 4 em relação ao grupo 3: uma vez esse último concluído com sucesso, habilita-se para o próximo grupo de subtarefas.

Para auxiliar o usuário a mantê-lo informado sobre essa e outras informações, utilizar-se-á na janela inferior um esquema de cores para mostrar o estado de cada tarefa. No início, todas as tarefas terão a cor branca - em contraste ao fundo escuro para ressaltá-las - indicando assim que ainda não foram realizadas. Ao selecionar uma determinada tarefa, essa mudará para a cor amarela, indicando que é a tarefa escolhida pelo usuário. Uma vez concluída com sucesso, passará para verde, caso contrário, terá sua cor mudada para vermelho. Caso uma tarefa seja escolhida, porém suas dependências não tenham sido previamente concluídas com sucesso, essas últimas terão o seu círculo piscando em vermelho (alternando com a cor de fundo). Essa relação é bastante simples, porém é uma informação de grande importância para o usuário, pois pode alertá-lo em tempo de prevenir problemas de usabilidade, como por exemplo desconforto com a aplicação e o uso de seus recursos, favorecendo assim o *controle do usuário*. O esquema de cores (tabela 2), bem como a relação de dependências, por sua relevância, deverão estar contidos no manual de ajuda da aplicação.

Cor	Significado
<i>Branco</i>	Tarefa não iniciada
<i>Amarelo</i>	Tarefa escolhida (em processo)
<i>Verde</i>	Tarefa concluída com êxito
<i>Vermelho</i>	Tarefa concluída sem êxito
<i>Vermelho intermitente</i>	Dependência não processada

Tabela 2: Relação das cores para as tarefas de configuração

A Moldura Superior

A moldura superior é responsável por apresentar as tarefas selecionadas na moldura inferior, além de informar o usuário sobre o andamento da ação como por exemplo tarefas não realizadas ou mal concluídas - através de mensagens de alerta. Portanto, para cada tarefa, a moldura é atualizada para apresentar apenas as informações cabíveis a cada uma das mesmas.

Assim que o usuário decide criar um novo *perfil*, o menu principal é sobreposto pela janela de criação. Na moldura inferior, todos os círculos estão inicialmente em branco (inclusive “Início” e “Fim”), e na moldura superior há as diretrizes para construir um novo perfil de forma simples e organizada, como por exemplo a escolha da ordem das tarefas e o botão “Fim”, o qual serve para finalizar a ação salvando ou descartando o *perfil* construído.

Na tarefa 1, é mostrada à direita uma área de desenho, cujas dimensões são proporcionais às de uma folha A4, e no fundo uma grade divisória, demarcando as células da matriz real. À esquerda, um conjunto de ferramentas de desenho e texto - semelhante a de aplicações conhecidas como o TuxPaint e o PaintBrush® - dá os instrumentos para criar as figuras representativas. Réguas em cima e à direita da área de desenho localizam as células, como indica a figura 31. Para melhorar o resultado final, a aplicação pode, mediante configuração do usuário, tentar deslocar as figuras para se enquadrarem em um conjunto mínimo de células (uso eficiente do espaço), bem como manter uma distância saudável entre as mesmas - em torno de 30 mm conforme (ISO, 1997) - prevenindo assim possíveis acessos indesejados (signos vizinhos) no futuro.

Na segunda tarefa, cujo propósito é relacionar cada um dos conjuntos de células a um signo, mantém-se a área de desenho, modificando apenas a caixa de ferramentas por um assistente de criação de signos - figura 32.

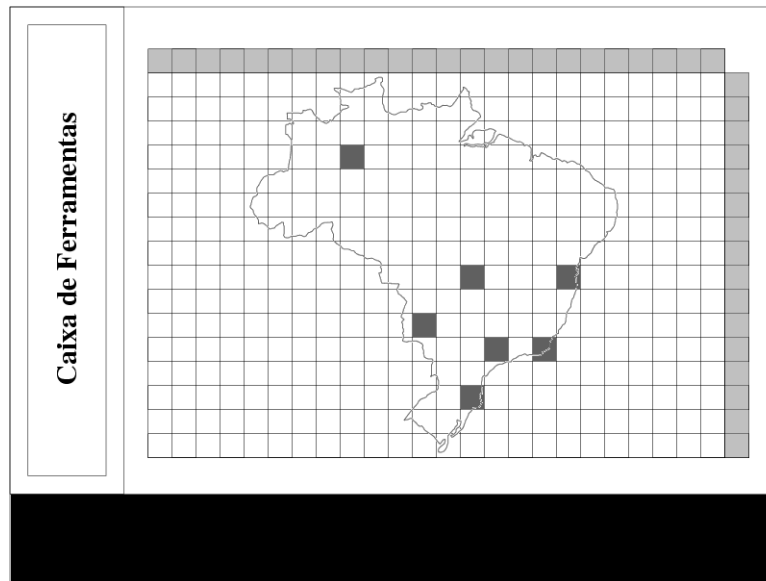


Figura 31: Desenho do mapa de signos

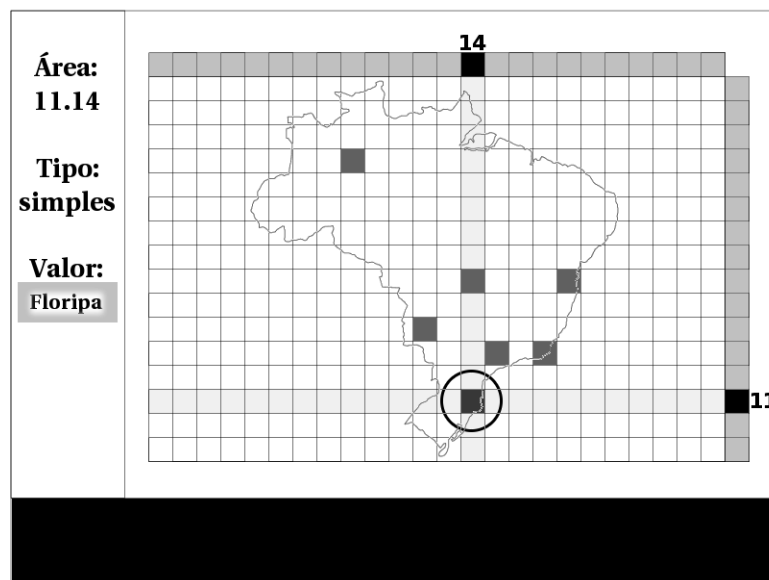






Figura 32: Atribuição dos signos

As outras duas tarefas, referentes à configuração dos módulos do *device driver*, estão descritas nas figuras 33 e 34. A primeira trata da seleção e ordenação dos módulos, onde constam duas colunas: a da esquerda lista todos os módulos instalados no sistema, porém não selecionados para o *perfil* (**Inativos**); a da direita lista todos os módulos já selecionados e que farão parte do futuro *perfil* (**Ativos**). Essas duas listas são mutuamente exclusivas; ou seja, não há a ocorrência de um mesmo item nas duas colunas (uma vez movido um módulo para uma das listas, esse é removido da outra). Essa movimentação é possível graças aos botões de direcionamento  e , os quais permitem mover um módulo de uma lista para a outra. Além disso, há outros dois botões de direcionamento no canto inferior direito:  e . Esses botões permitem ordenar os módulos previamente selecionados. Isso garante que toda ação feita através do *mouse* poderá ser feita via teclado (premissa de usabilidade).

A figura 34 ilustra a configuração de cada módulo que outrora havia sido escolhido na tarefa anterior (lista **Ativos**). Como cada módulo tem a sua própria configuração, cabe à aplicação base apenas fornecer uma API para a incorporação de interfaces provindas dos módulos adicionais.

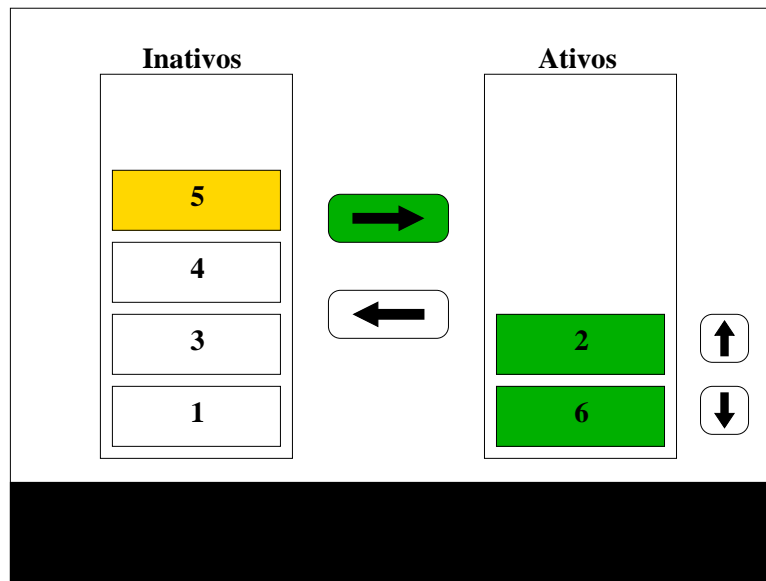


Figura 33: Seleção e ordenação dos módulos

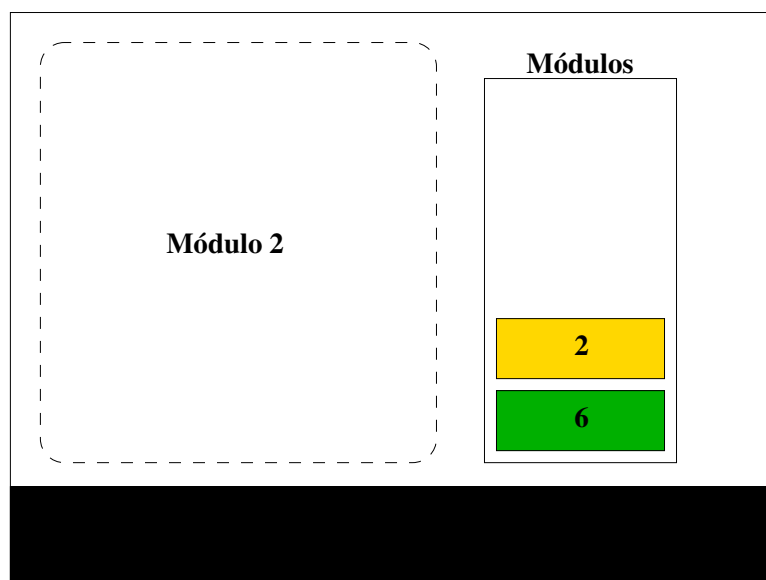


Figura 34: Configuração extra dos módulos

Por último, cabe ao usuário realizar a tarefa *Fim*, onde será decidido se o *perfil* será ou não armazenado no sistema (figura 35). Cabe ressaltar que o padrão de cores adotado na moldura inferior se mantém em todas as tarefas.

Se o usuário decidir armazenar o *perfil* construído, ele deverá informar um frase-lembrete para discriminá-lo no sistema. Internamente, o programa gerará uma assinatura (*hash*) desse *perfil* para armazená-lo em um banco de dados, cuja recuperação será possível através da frase-lembrete e de uma miniatura do mapa de signos (*thumbnail*), conforme será visto adiante.

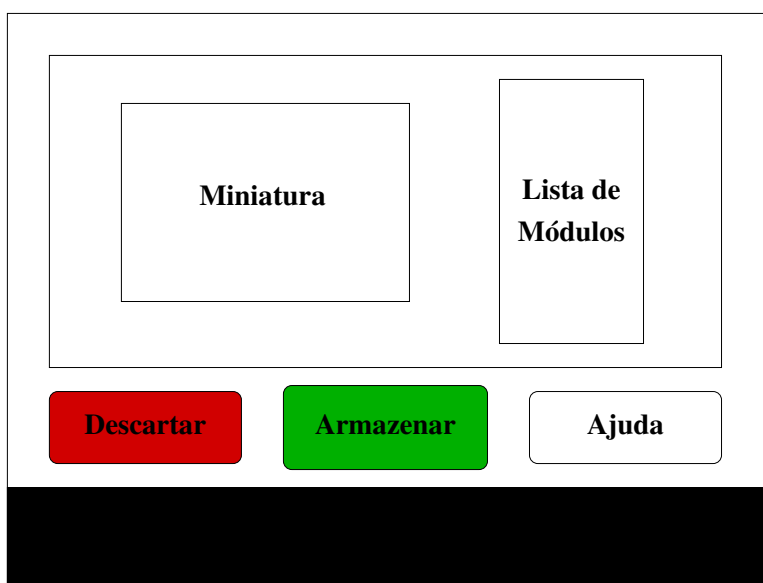


Figura 35: Armazenar ou descartar um *perfil*

6.1.2 Janela de Depósito de Perfis

Uma vez armazenados os *perfis* no sistema, esses podem ser recuperados a qualquer momento por meio da janela de depósito de *perfis*. Nessa janela, o usuário pode selecionar um dentre os vários *perfis* que se encontram no sistema (já catalogados) ou optar por recuperá-lo a partir de um arquivo. Esse último recurso, em especial, permitirá a portabilidade do *perfil* para diferentes computadores e Sistemas Operacionais. A figura 36 ilustra a janela de depósito com as suas opções.

Como pode-se perceber, há uma lista de *perfis* cujas indicações se dão por dois atri-

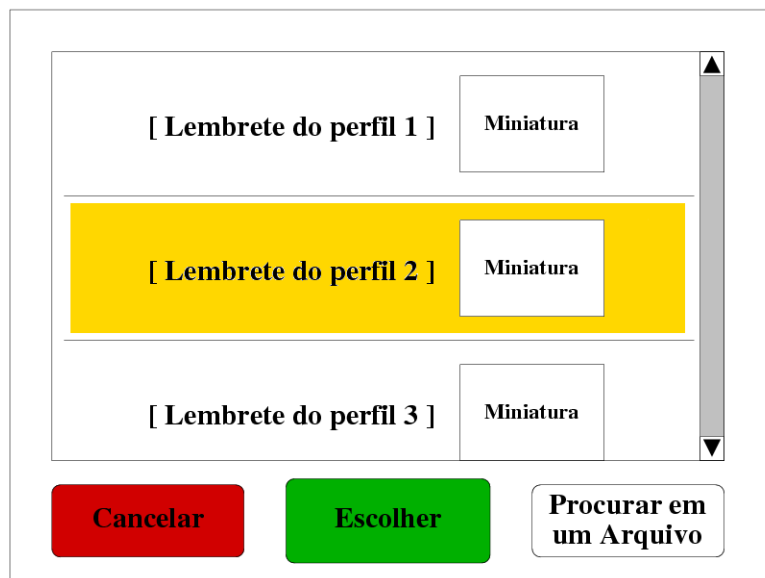


Figura 36: Submenu de recuperação de um perfil

butos: uma frase-lembrete² e uma miniatura visual do mapa de signos. Pode-se inclusive, utilizando a opção **Procurar em um Arquivo**, adicionar outros *perfis* que se encontram em arquivos externos à *COLMÉIA*. O esquema de cores se mantém fiel ao das outras janelas.

Uma vez selecionado um *perfil*, é apresentado na tela o submenu de ativação, conforme a figura 37. A partir desse momento, pode-se ativar o *perfil* selecionado para, então, utilizá-lo prontamente - opção **Ativar o perfil**. Nesse caso, será aconselhável, porém não obrigatório, que o usuário insira o mapa de signos correspondente no dispositivo físico, a fim de manter as correspondências entre células e signos.

A segunda opção, **Alterar o perfil**, apresentará a janela de criação de *perfis* para realizar alterações no *perfil* escolhido, como por exemplo adicionar novos signos ao mapa. As outras duas opções, **Escolher outro perfil** e **Voltar ao menu principal** permitem que o usuário, a qualquer momento, possa navegar por toda a aplicação.

Na figura 38 estão listadas todas as possibilidades de navegação da aplicação base. A partir do menu principal, é possível deslocar-se para a janela de criação de *perfis* (ação 1) ou para a janela de depósito de *perfis* (3). Uma vez concluída a criação, volta-se ao menu principal (2) - em condições normais. O mesmo acontece na recuperação de um

²Procura-se diminuir a carga do usuário utilizando frases-lembrete ao invés de uma estrutura de arquivos e diretórios.

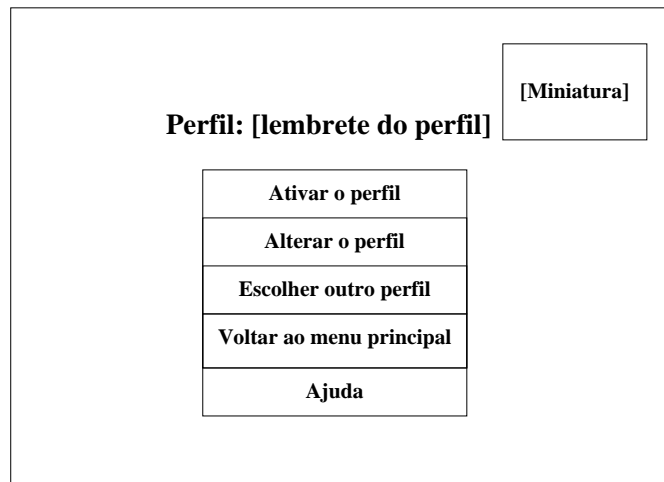


Figura 37: Submenu de ativação de um perfil

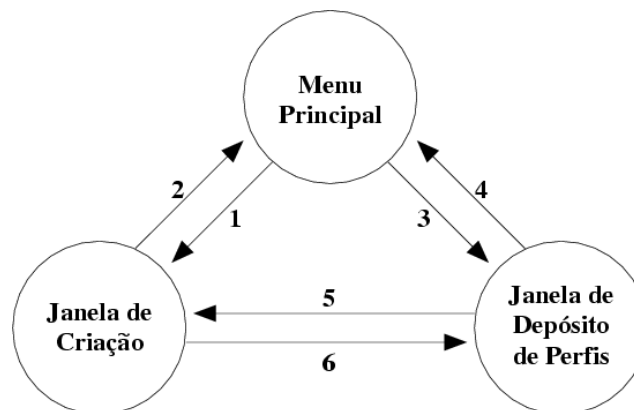


Figura 38: Possibilidades de navegação da aplicação base

perfil: concluída essa, pode-se voltar ao menu principal (4). Além disso, pode-se alterar um *perfil* a partir da janela de depósito (5), e somente nesse caso é que pode-se mover da janela de criação para a de depósito (6).

6.2 Considerações Finais

Além dos aspectos referentes à descrição da interface mencionados ao longo do capítulo, há alguns itens adicionais que podem melhorar a interação homem-máquina, seja na criação de um *perfil* ou mesmo na sua recuperação:

- A transição entre as janelas e menus pode ser feita utilizando efeitos de animação, apresentando assim ao usuário todas a(s) consequência(s) de uma ação por ele tomada (*retorno*). Sombreamento e deslocamento dos menus são alguns exemplos;
- Para toda ação oferecida ao usuário, deve haver uma tecla de atalho para que a mesma possa ser realizada sem o uso de *mouse* ou outro objeto apontador. Para tanto, deve-se recorrer a atalhos conhecidos (Ctrl+C, Ctrl+V, etc.) e/ou de fácil assimilação: Ctrl+D para deslocar um módulo da coluna da esquerda para a direita, por exemplo. A ação do tipo arrastar-e-soltar (*drag'n'drop*), comum nas aplicações gráficas, deve permanecer presente;
- Como complemento ao item anterior, utilizar balões explicativos. O balão deve conter uma explicação bastante sintética da ação, bem como a sua tecla de atalho. Esse recurso foi pensando para que possa, futuramente, ser lido por meio de um sintetizador digital que interprete a aplicação, assim como faz a aplicação Gnopernicus no ambiente GNOME (THE GNOME PROJECT, 2003a);
- Nas janelas onde são apresentadas informações novas ao usuário (termos ou ações), deverá haver uma opção de ajuda contendo as devidas explicações para contextualizar o usuário;
- Apresentar poucas informações por tela. Isso previne a sobrecarga do usuário ao assimilar e correlacionar todos os dados reunidos, como descreve (GONÇALVES et al., 2002, p. 2091).

Esses recursos extras, imersos dentro da aplicação, servirão para manter o usuário bem informado - sem exageros - e no controle da situação. As explicações adicionais, balões e ajuda, serão visualizadas somente a partir de uma ação do usuário, reforçando as premissas de *controle do usuário*, de *brevidade* e de *retorno*) e aplicadas na definição das telas e suas relações.

7 *Discussões e Conclusões*

O desenvolvimento da *COLMÉIA* partiu da hipótese inicial de beneficiar uma classe reduzida de portadores de necessidades especiais: os portadores de deficiência motora cujos movimentos dos membros superiores se encontram comprometidos. Essa decisão refletiu diretamente em cada uma das etapas de desenvolvimento da tecnologia, em especial a escolha do método de ativação, baseado em infravermelho, e a definição dos requisitos para a definição da *COLMÉIA* como alternativa de entrada de dados.

Entretanto, durante o processo de descrição da tecnologia, pôde-se perceber que a *COLMÉIA* possui um espectro de atuação mais amplo que o previsto. Outros casos, como por exemplo a deficiência visual, também podem ser contemplados pela tecnologia. Para esse exemplo, há a possibilidade de utilizar mapa de signos em Braille: uma folha, contendo os signos dessa linguagem representados por perfurações ou em alto-relevo, pode ser sobreposta à tela de retorno tátil - esta rígida - e utilizada sem perda qualitativa de acessibilidade. Na parte do *device driver*, agrega-se à *COLMÉIA* um módulo de acesso temporal, onde o sistema será capaz de diferenciar entre a leitura e o acesso real a um símbolo específico, utilizando para tal o tempo como fator limite¹, abrangendo assim os aspectos sensoriais e “comportamentais” da entrada de dados.

Muitas outras idéias podem ser concebidas e implementadas, pois este é o propósito dessa tecnologia: prover recursos (multimídia) para a criatividade, promovendo a curiosidade e a auto-expressão, assim como pregava (PAPERT, 1994). Tais qualidades tendem tão somente a melhorar o rendimento dos aprendizes, retirando-os de sua posição passiva (bancária) e colocando-os como agentes ativos da sua própria educação, como afirma (FREIRE, 1987, p. 70):

¹Esse assunto já foi abordado na página 53.

“Assim é que, enquanto a prática bancária, como enfatizamos, implica uma espécie de anestesia, inibindo o poder criador dos educandos, a educação problematizadora, de caráter autenticamente reflexivo, implica um constante ato de desvelamento da realidade. A primeira pretende manter a *imersão*; a segunda, pelo contrário, busca a *emersão* das consciências, de que resulte sua *inserção crítica* da realidade.”

Entretanto, não foi objetivo deste trabalho construir modelos pedagógicos onde o computador fosse participante do processo, mas sim ampliar as capacidades das máquinas para que sirvam como um real instrumento a serviço do educador. Embora as condições de trabalho desse profissional não estejam ainda em um grau satisfatório, como vem destacando a mídia nacional nesses últimos anos (desde 1998² até os dias atuais³), políticas federais têm procurado mudar esse panorama, pelo menos no aparelhamento básico: computadores. Em 2004, o Programa Nacional de Informática na Educação (PROINFO) pretende distribuir 5.000 computadores para as escolas públicas, totalizando mais de 53.000 unidades distribuídas ao longo de sete anos⁴. Além disso, o governo federal criou, nesse ano de 2004, um programa de doação de computadores usados, denominado *Computadores para Inclusão*⁵, cujo valor pode chegar a 120.000 computadores destinados às instituições públicas, entre elas escolas e centros comunitários. Como pode-se notar, o parque tecnológico das instituições públicas de ensino tem apresentado uma melhora sensível, porém significativa, fazendo com que hoje se dia se possa pensar em aulas e laboratórios de informática nas escolas públicas.

Embora a inserção da informática não se resuma simplesmente à instalação de computadores nas escolas, essa é necessariamente uma de uma série de etapas para se atingir o objetivo final. Concomitantemente, deve haver uma clara política de capacitação profissional e formação de agentes multiplicadores, os quais serão os responsáveis por introduzir - de fato - a informática na educação como um facilitador na construção do conhecimento.

Mais especificamente, em relação ao uso da Informática na educação especial, há também o estabelecimento de uma política pública, com destaque ao Projeto de Informática na Educação Especial (BRASIL, 2000). Sob tais bases, prevê-se o computador não apenas como um instrumento de auxílio no processo de ensino-aprendizagem, mas sobretudo uma tecnologia para a promoção da autonomia do portador de necessidades es-

²FOLHA DE S. PAULO. *País tem 21% de professores sem formação mínima*. São Paulo. 11/02/1998. Caderno 3. p. 8.

³ZERO HORA. *Um em cada três professores é pobre*. Porto Alegre. 25/05/2004.

⁴AGÊNCIA ESTADO. *MEC instalará mais 5 mil PCs em escolas públicas*. 06/05/2004.

⁵AGÊNCIA BRASIL. *Governo cria programa de doação de computadores usados*. 25/05/2004.

peciais no ambiente educacional. Contudo, tal decisão implica em uma nova problemática quando se imagina a tecnologia assistiva computacional em um uso mais universal: compensar as limitações dos portadores de necessidades especiais, sem com isso comprometer o rendimento dos não portadores. Tecnologias como a **COLMÉIA**, dadas as suas características adaptativas, podem ser aplicadas em tais ambientes, no anseio de prover um espaço educacional único onde todos poderão utilizar um mesmo dispositivo e, por conseguinte, amenizar as diferenças que podem representar barreiras sociais, como afirma (SANTOS, 2002):

“Temos o direito de ser iguais quando a diferença nos inferioriza e de ser diferentes quando a igualdade nos descaracteriza.”

Em alguns casos, como por exemplo (DAMASCENO; FILHO, 2003), a necessidade de uma tecnologia flexível é ainda mais perceptível, dadas as necessidades de cada usuário, seja em termos de aspectos sensoriais e/ou “comportamentais”. Nesse exemplo, percebeu-se a necessidade de uma tecnologia que atendesse aos requisitos de *força*, *alteração do mapa de signos* e *signo composto*⁶, assim como faz a **COLMÉIA** e soluções já conhecidas e amplamente utilizadas, como por exemplo os teclados virtuais.

Portanto, a utilização de tais tecnologias se dá no momento em que se percebe a sua real necessidade nos ambientes de ensino. Bem aplicadas, essas tecnologias poderão - de fato - auxiliar a integração dos alunos em prol de uma educação mais democrática e igualitária - sem desconsiderar a personalidade de cada um. E nesse contexto cabe ao professor o papel de facilitador e condutor da prática educativa, utilizando para isso ferramentas que facilitem o processo, com destaque às tecnologias assistivas e complementares computacionais.

7.1 Resultados Obtidos

Considerando todos os aspectos relacionados na seção anterior, a descrição de uma tecnologia como a **COLMÉIA** mostrou-se de grande relevância, pois a sua base foi calçada nas tecnologias tradicionais, mas os seus limites se equiparam às tecnologias mais inovadoras, dada a combinação híbrida de interoperabilidade e funcionalidades. A pesquisa inicial (o levantamento do estado da arte) foi fator decisivo para todo o processo. Com base na fundamentação inicial, produziu-se uma síntese das tecnologias de entrada

⁶Requisitos vistos na página 37.

de dados que permitiu identificar os requisitos desejáveis à *COLMÉIA*. O confronto desses requisitos com o conjunto das soluções já existentes resultou no subconjunto das bases tecnológicas que atendessem aos interesses do periférico, do *device driver* e da aplicação base e dos seus modos de funcionamento e tratamento da informação (interna e externamente entre as camadas). A linguagem de interação homem-máquina, subproduto desse último objetivo de descrição, também foi concebida e estabelecida para garantir a gama de possibilidades previstas na *COLMÉIA*.

Para concluir, cabe destacar que apenas o último objetivo específico não foi alcançado: a análise de aplicabilidade. Uma primeira estimativa de levantamento de custos em componentes eletroeletrônicos para a produção de um protótipo estabeleceu o valor de R\$150,00. Contudo, todo o processo de prototipação (desenvolvimento de *hardware* e *software*) e realização do estudo de caso chegaria a números mais expressivos, o que tornou a prática inviável até a conclusão deste texto. Fica, portanto, o estudo de caso em aberto para uma futura continuação do trabalho.

Ao final deste projeto, crê-se que foi preenchida uma lacuna temporal entre as tecnologias vigentes e aquelas consideradas como vanguarda, onde destacam-se os reconhecimentos de voz e de movimentos (pela naturalidade com que podem ser utilizadas, pois procuram na comunicação humana a base para a interação com a máquina). Se por um lado as primeiras são de fácil alcance e demonstram insuficiência de recursos, por outro lado as últimas ainda não alcançaram a maturidade ou condições favoráveis (custo, assimilação, etc.) para a sua utilização pela grande maioria das pessoas - sejam essas portadores de necessidades especiais ou não. Esse momento de transição entre os dois paradigmas está sendo, atualmente, representado e gerido por tecnologias mais flexíveis, como por exemplo as telas sensíveis ao toque e os teclados de conceitos. A esse último grupo enquadra-se a *COLMÉIA*, uma tecnologia voltada ao ser humano, pensada no ser humano, fazendo do computador um canal próspero e fecundo à promoção da educação de todos.

7.2 Trabalhos Futuros

O próximo passo vislumbrado para a continuação do projeto é, sem dúvida, realizar a análise de aplicabilidade da tecnologia, visto que foi esse o ponto esperado e faltante deste trabalho. Assim como acontece com a educação regular, tem-se procurado propiciar

espaços beneficiados pela Informática e seus recursos (ACS, 2004). A Fundação Catarinense de Educação Especial (FCEE, 2002), órgão responsável pela definição e execução da política de educação especial no Estado, por sua história e trabalho reconhecidos na região, seria um dos locais indicados para a realização do estudo de caso.

Outros trabalhos, mais voltados ao desenvolvimento tecnológico, são o de construir módulos adicionais do *device driver*, uma vez que a **COLMÉIA** abre essa possibilidade de agregar novas funcionalidades. Módulos de correção de acesso por tempo e de reconhecimento de movimentos, por exemplo, são exemplos de contribuições que multiplicarão as potencialidades de uso da tecnologia. E esse trabalho pode seguir além: pode-se construir um servidor WEB de *perfis* de usuários, onde poderão ser catalogados os *perfis* que se destinam a um uso específico ou necessidade especial.

Referências

3M DYNAPRO. *Near Field Imaging - Touch Screen System*. 3. ed. [S.l.], maio 2001.

AGÊNCIA BRASIL. *Governo cria programa de doação de computadores usados*. 25/05/2004.

AGÊNCIA ESTADO. *MEC instalará mais 5 mil PCs em escolas públicas*. 06/05/2004.

AGÊNCIA ESTADO. *Cresce a inclusão de alunos especiais em escolas regulares*. setembro 2003.

ÁREA TÉCNICA DE SAÚDE DO TRABALHO. *Lesões Por Esforços Repetitivos (LER) / Distúrbios Osteomusculares Relacionados ao Trabalho (DORT)*. Brasília: Ministério da Saúde, 2001. (A - Normas e Manuais Técnicos, v. 103).

ASSESSORIA DE COMUNICAÇÃO SOCIAL. *Educação especial ganha cem novos laboratórios*. julho 2004. Disponível em: <<http://www.mec.gov.br/acs/asp/noticias/noticiasDiaImp.asp?id=6425>>.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *Norma Técnica NBR 9241-11: Requisitos ergonômicos para trabalho de escritórios com computadores - parte 11 - orientações sobre usabilidade*. Rio de Janeiro, 2000. 21 p.

ATD. *Object-Oriented Programming with X-Designer*. 2004. Disponível em: <<http://rd13doc.cern.ch/Atlas/Notes/004/Note004-7.html>>.

BENSON, C. et al. *GNOME Human Interface Guidelines*. [S.l.], 2002.

BEZERRA, C. M. C. *BR Braille: programa tradutor de textos braille digitalizados para caracteres alfanuméricos em português*. Campinas, 2003.

BIOMEDICAL, INDUSTRIAL AND HUMAN FACTORS ENGINEERING. *Rehabilitation Engineering*. 2004. Disponível em: <<http://www.cs.wright.edu/bie/rehabengr/>>.

BRASIL. *Lei Federal nº 7.853 de 24 de outubro de 1989*. Brasília: [s.n.], 1989.

BRASIL. *Lei Federal nº 8.069 de 13 de julho de 1990: Estatuto da criança e do adolescente*. Brasília: [s.n.], 1990.

BRASIL. *Lei Federal nº 9.394 de 20 de setembro de 1996: Estabelece as diretrizes e bases da educação nacional*. Brasília: [s.n.], 1996.

BRASIL. *Parâmetros Curriculares Nacionais: Adaptações curriculares*. Brasília: MEC / SEF / SEESP, 1998. 62 p.

BRASIL. *PROINESP - Projeto de Informática na Educação Especial*. Brasília: MEC / SEESP, 2000.

BRASIL. *Diretrizes nacionais para a educação especial na educação básica*. Brasília: MEC / SEESP, 2001. 79 p.

BRASIL. *Censo Escolar 2002*. Brasília: MEC, 2003.

BRASIL. *Censo Escolar 2003*. Brasília: MEC, 2004.

BROUWER, A. *The Linux keyboard and console HOWTO*. 2.13. ed. [S.l.], outubro 2002. Disponível em: <<http://www.tldp.org/HOWTO/Keyboard-and-Console-HOWTO.html>>.

CAMPOS, M. de B.; SILVEIRA, M. S. *Tecnologias para educação especial. IV Congresso RIBIE, Brasília, 1998*.

CANESTA. *Canesta Keyboard*. 2002. Disponível em: <<http://www.canesta.com/products.htm>>.

CHEN, T. T. H.; FELLS, S.; MIN, S. S. *Flowfield and beyond: applying pressure-sensitive multi-point touchpad interaction*. In: . Baltimore, EUA: IEEE International Conference on Multimedia & Expo, 2003.

CLIK. *Clik Tecnologia Assistiva*. 2002. Disponível em: <<http://www.clik.com.br>>.

COMPLEX. *Everest*. 2003. Disponível em: <<http://www.complex.com.br>>.

CONSELHO NACIONAL DE EDUCAÇÃO. *CÂMARA DE EDUCAÇÃO BÁSICA. Resolução CNE/CEB nº 2 de 11 de setembro de 2001*. Brasília, 2001.

COX, P. *Math and Music: a primer*. 2004. Disponível em: <<http://members.cox.net/mathmistakes/music.htm>>.

CYBIS, W. de A. *Engenharia de Usabilidade: Uma abordagem ergonômica*. Florianópolis, maio 2003.

DAMASCENO, L. L.; FILHO, T. A. G. ao. *As Novas Tecnologias como Tecnologia Assistiva: Usando os recursos de acessibilidade na educação especial*. 2003. Disponível em: <<http://www.proinfo.mec.gov.br/biblioteca/textos/Recursos.pdf>>.

DATAFOLHA INSTITUTO DE PESQUISAS. *Prevenção às LER/DORT: Sintomas*. São Paulo, SP, outubro 2001.

FCEE. *Fundação Catarinense de Educação Especial*. 2002. Disponível em: <<http://www.sc.gov.br/webfcee>>.

FINGERWORKS, INC. *FingerWorks*. 2003. Disponível em: <<http://www.fingerworks.com>>.

FOLHA DE S. PAULO. *País tem 21% de professores sem formação mínima*. São Paulo. 11/02/1998. Caderno 3. p. 8.

- FREIRE, P. *Pedagogia do Oprimido*. 17. ed. Rio de Janeiro: Paz e Terra, 1987.
- GONÇALVES, C. E. et al. Tadeu - a different approach to on screen keyboard systems. *XIV Congresso Brasileiro de Automática*, 2002.
- INTELLITOOLS. *Intellitools, Serving the Diverse Classroom for Over 20 years*. 2003. Disponível em: <<http://www.intellitools.com>>.
- INTERNATIONAL STANDARDS ORGANIZATION. *International Standard 9421-14: Ergonomic requirements for office work with visual display terminals (vdt) - part 14: menu dialogues*. Suíça, 1997. 57 p.
- LAGARES, J. *Plaphoons*. 2004. Disponível em: <<http://www.lagares.org>>.
- LEGO.COM. *LEGO Mindstorms*. 2003. Disponível em: <<http://mindstorms.lego.com>>.
- MIRABILIS. *ICQ*. 2004. Disponível em: <<http://www.icq.com>>.
- OLIVEIRA, D. de et al. *Inclusão digital de portadores de necessidades especiais através da comunicação móvel*. 2004. Disponível em: <<http://www.cts.org.br/rybena.htm>>.
- ORSO, P. J. et al. *Reflexões acerca das Lesões por Esforços Repetitivos e a Organização do Trabalho*. Campinas, SP: [s.n.], 2001.
- PAPERT, S. *A Máquina das Crianças: Repensando a escola na era da informática*. Porto Alegre: Artes Médicas, 1994.
- PROJETO SOFTWARE LIVRE BRASIL. *5º FISL: É possível escrever com os olhos, com Software Livre*. maio 2004. Disponível em: <<http://www.softwarelivre.org/news/2276>>.
- REDE SACI. *Kit Saci 2*. 2004. Disponível em: <<http://www.saci.org.br>>.
- REMINGTON PRODUCTS COMPANY. *Remington History*. 2002. Disponível em: <<http://www.remington-products.com/australia/corpinfo/history.html>>.
- SANCHES, A. L. G. *Programação em GTK e C++*. 2003. Disponível em: <<http://sl.geness.ufsc.br>>.
- SANTAROSA, L. M. C. Adaptação para o português e avaliação de um simulador de teclado para portadores de paralisia cerebral. In: . [S.l.]: Anais do II Congresso Ibero-Americano de Informática Educativa, 1994. v. 1.
- SANTOS, B. de S. *A Globalização e as Ciências Sociais*. São Paulo: Cortez, 2002.
- SCHMIDT, S. *FreeBSD Kernel Interfaces Manual: Keyboard(4)*. [S.l.], janeiro 1995.
- SINDICATO DOS TRABALHADORES NO PODER JUDICIÁRIO FEDERAL EM SANTA CATARINA. *L.E.R. - Lesões por Esforços Repetitivos: Você não pode ser mais uma vítima*. Florianópolis, SC, 1998.
- TANENBAUM, A. S. *Modern Operating Systems*. [S.l.]: Prentice-Hall, 1992.

THE GNOME PROJECT. *GNOME*. 2003. Disponível em: <<http://www.gnome.org>>.

THE GNOME PROJECT. *GOK*. 2003. Disponível em:
<<http://developer.gnome.org/projects/gap/AT/GOK>>.

UFRJ. *Projeto DOSVOX*. 2002. Disponível em: <<http://intervox.nce.ufrj.br/dosvox>>.

UNESCO. *Declaração de Salamanca e Linhas de Ação sobre Necessidades Educativas Especiais*. Brasília, DF: CORDE, 1994.

WIKIPEDIA. *Computer Mouse - Wikipedia*. abril 2003. Disponível em:
<http://www.wikipedia.org/wiki/Computer_mouse>.

ZELENOVSKY, R.; MENDONÇA, A. *PC: Um guia prático de hardware e interfaceamento*. Rio de Janeiro: MZ, 1999.

ZERO HORA. *Um em cada três professores é pobre*. Porto Alegre. 25/05/2004.

ANEXO A – Frequências das Notas Musicais

A	55	110	220	440	880	1760	3520	7040
A#	58,2	116,5	233	466,1	932,3	1864,6	3729,3	7458,6
B	61,6	123,4	246,8	493,8	987,7	1975,4	3951	7902,1
C	65,2	130,7	261,4	523,1	1046,4	2092,8	4185,9	8371,9
C#	69	138,4	276,9	554,2	1108,6	2217,2	4434,8	8869,7
D	73,1	146,6	293,3	587,1	1174,5	2349	4698,5	9397,1
D#	77,4	155,3	310,7	622	1244,3	2488,6	4977,8	9955,8
E	82	164,5	329,1	658,9	1318,2	2636,5	5273,7	10547,8
F	86,8	174,2	348,6	698	1396,5	2793,2	5587,2	11175
F#	91,9	184,5	369,3	739,5	1479,5	2959,2	5919,4	11839,5
G	97,3	195,4	391,2	783,4	1567,4	3135,1	6271,3	12543,5
G#	103	207	414,4	829,9	1660,6	3321,5	6644,2	13289,3

Tabela 3: Frequências das notas musicais de 8 oitavas da escala igualmente temperada.

Valores em Hertz aproximados com 1 dígito de precisão (COX, 2004).

ANEXO B – Lista de Ações Reconhecidas pelo Device Driver

Tipo de Ação	Descrição	Sintaxe da Ação
Edição	Copia a área de seleção	copia
	Cola	cola
	Desfaz a última ação	desfaz
	Recorta a área de seleção	recorta
	Seleciona todo o documento	seleciona tudo
Imagem	Captura a imagem da janela focada	captura janela
	Captura a imagem da tela	captura tela
Janela	Encerra a aplicação focada	fecha
	Maximiza a janela focada	maximiza
	Minimiza a janela focada	minimiza
	Move a janela focada	move
Sistema	Desliga o computador	desliga
	Executa uma aplicação	executa <comando>
	Reinicia o computador	reinicia

Tabela 4: Ações reconhecidas pelo *device driver* e remetidas ao SO